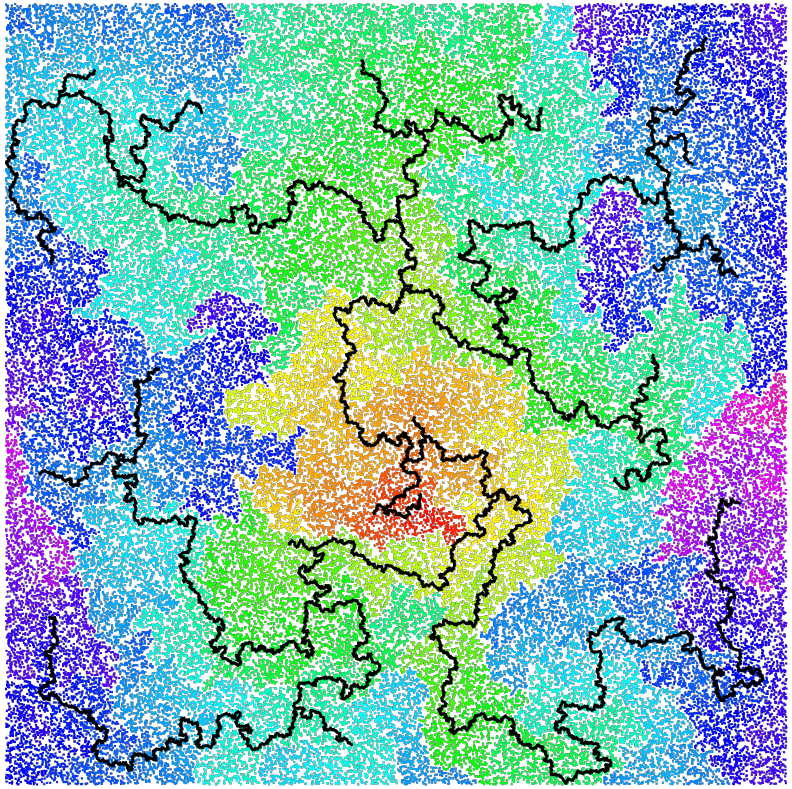
The scaling limit of the minimum spanning tree, and the critical Erdős-Rényi random graph



L. Addario-Berry

Summer school on random graphs and probabilistic methods

Fields Institute

June 9, 2017

Parts are joint with Nicolas Broutin, Christina Goldschmidt, Grégory Miermont

Percolation on the complete graph

- Complete graph K_n: vertices $1, \ldots, n$ and all possible edges, independent uniform $[0, 1]$ edge weights

Percolation on the complete graph

- Complete graph K_n: vertices $1, \ldots, n$ and all possible edges, independent uniform $[0, 1]$ edge weights
- Denote by $G_{n,p}$ the graph obtained by only keeping edges of K_n of weight less than p.

Percolation on the complete graph

- Complete graph K_n: vertices $1, \ldots, n$ and all possible edges, independent uniform $[0, 1]$ edge weights
- Denote by $G_{n,p}$ the graph obtained by only keeping edges of K_n of weight less than p.
- In $G_{n,p}$, each edge is independently present with probability p.

Percolation on the complete graph

- Complete graph K_n: vertices $1, \ldots, n$ and all possible edges, independent uniform $[0, 1]$ edge weights
- Denote by $G_{n,p}$ the graph obtained by only keeping edges of K_n of weight less than p.
- In $G_{n,p}$, each edge is independently present with probability p.
- Write T_n for the MST of K_n with these weights, $T_{n,p}$ for the graph obtained by only keeping edges of T_n of weight less than p.

Global structure

$\star$ Set-up: K_n, edge weights $\{W_e, e \in E(K_n)\}$ are iid, Uniform$[0,1]$ (though the precise distribution is unimportant).

Global structure

$\star$ Set-up: K_n, edge weights $\{W_e, e \in E(K_n)\}$ are iid, Uniform$[0,1]$ (though the precise distribution is unimportant).
$\star$ Rescale the resulting MST so each edge has length $n^{-1/3}$, and give each vertex mass $1/n$.

Global structure

$\star$ Set-up: K_n, edge weights $\{W_e, e \in E(K_n)\}$ are iid, Uniform$[0,1]$ (though the precise distribution is unimportant).
$\star$ Rescale the resulting MST so each edge has length $n^{-1/3}$, and give each vertex mass $1/n$. Call the result $\mathcal{T}_n$.

Global structure

$\star$ Set-up: K_n, edge weights $\{W_e, e \in E(K_n)\}$ are iid, Uniform$[0,1]$ (though the precise distribution is unimportant).
$\star$ Rescale the resulting MST so each edge has length $n^{-1/3}$, and give each vertex mass $1/n$. Call the result $\mathcal{T}_n$.

Theorem (A-B, Broutin, Goldschmidt, Miermont 2013)

As $n \to \infty$, $\mathcal{T}_n \xrightarrow{\mathrm{d}} \mathcal{M}$, for some random measured metric space $\mathcal{M}$, in the Gromov–Hausdorff–Prokhorov sense.

Global structure

$\star$ Set-up: K_n, edge weights $\{W_e, e \in E(K_n)\}$ are iid, Uniform$[0,1]$ (though the precise distribution is unimportant).
$\star$ Rescale the resulting MST so each edge has length $n^{-1/3}$, and give each vertex mass $1/n$. Call the result $\mathcal{T}_n$.

Theorem (A-B, Broutin, Goldschmidt, Miermont 2013)

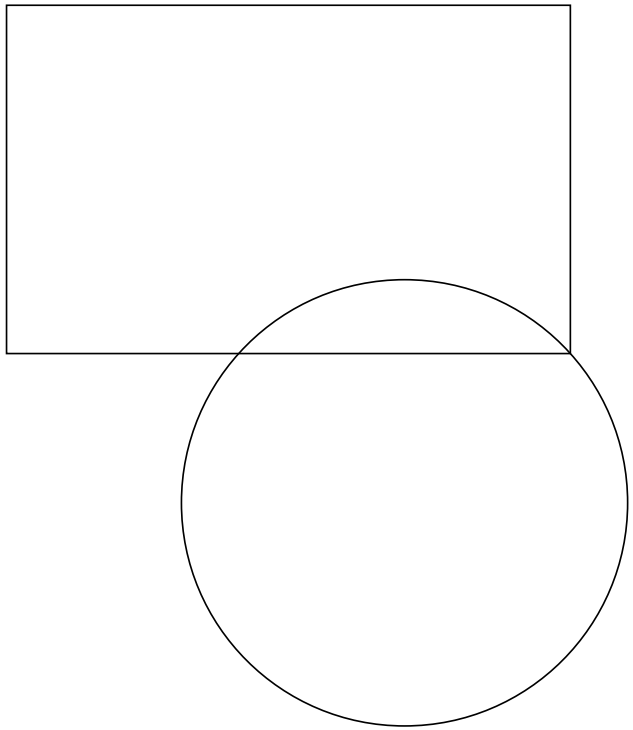
As $n \to \infty$, $\mathcal{T}_n \xrightarrow{\mathrm{d}} \mathcal{M}$, for some random measured metric space $\mathcal{M}$, in the Gromov–Hausdorff–Prokhorov sense.

We can also say some things about the limit $\mathcal{M}$; but first things first – a brief explanation of the notion of convergence.

Metric space convergence.

Given sets S, T in a metric space (M, d), let

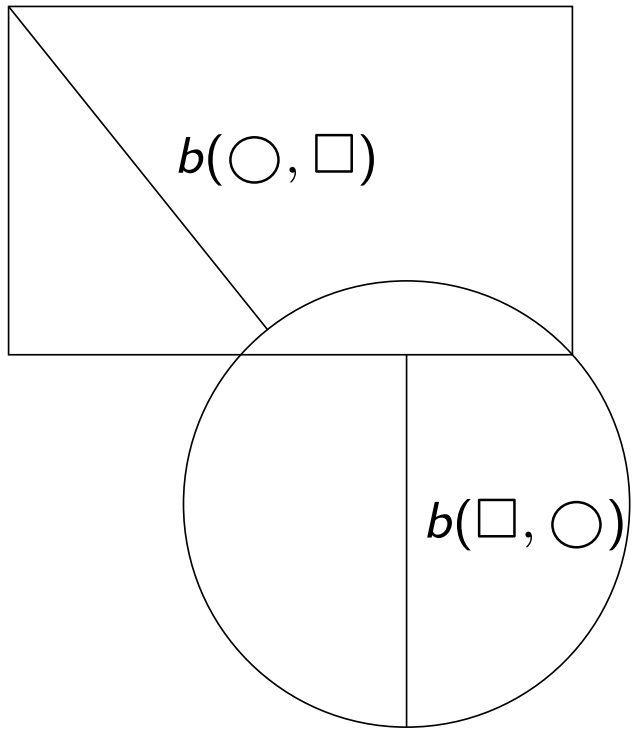
$$b(S, T) = \inf\{r \ : \ T \subset b_r(S)\}.$$



Metric space convergence.

Given sets S, T in a metric space (M, d), let

$$b(S, T) = \inf\{r \ : \ T \subset b_r(S)\}.$$



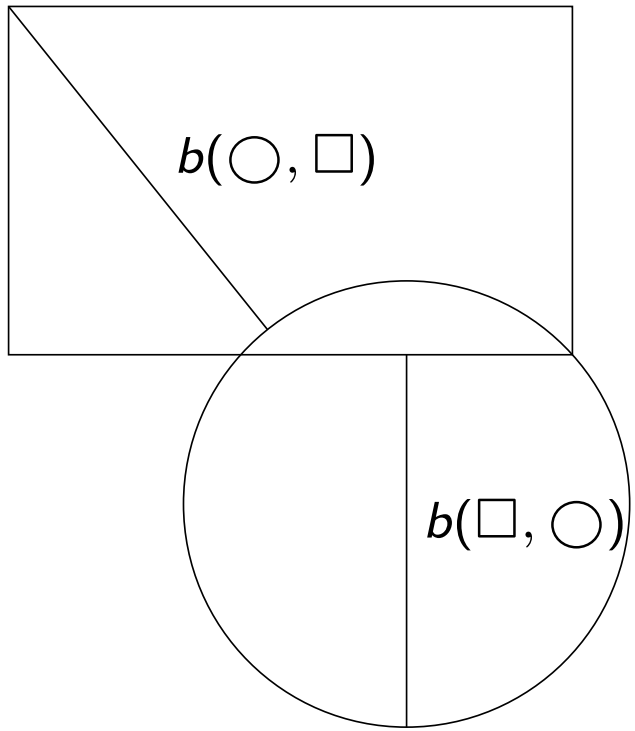
Metric space convergence.

Given sets S, T in a metric space (M, d), let

$$b(S, T) = \inf\{r \ : \ T \subset b_r(S)\}.$$

The *Hausdorff distance*

$d_H(S, T)$ between S and T is $\max(b(S, T), b(T, S))$.



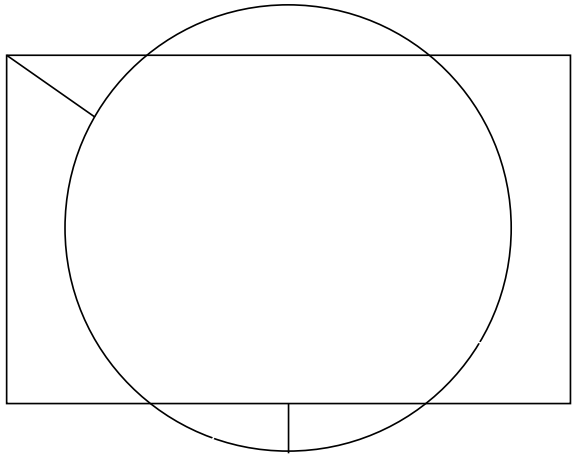
Metric space convergence.

Given sets S, T in a metric space (M, d), let

$$b(S, T) = \inf\{r \ : \ T \subset b_r(S)\}.$$

The *Hausdorff distance*

$d_H(S, T)$ between S and T is $\max(b(S, T), b(T, S))$.

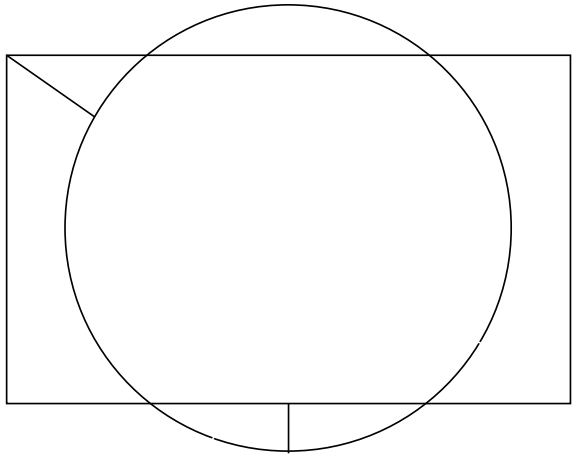


Metric space convergence.

Given sets S, T in a metric space (M, d), let

$$b(S, T) = \inf\{r \ : \ T \subset b_r(S)\}.$$

The *Hausdorff distance* $d_H(S, T)$ between S and T is $\max(b(S, T), b(T, S))$.



The *Gromov-Hausdorff* distance between metric spaces (S, d_1) and (T, d_2) is

$$\inf\{d_H(S, T)\},$$

where the infimum is over all metric spaces (M, d) containing both $(S, d_1$ and (T, d_2) as subspaces.

Similarity of metric spaces

Two metric spaces (X, d_X) and (Y, d_Y) are similar in the Gromov-Hausdorff Sense if we can map both into a common space (Z, d_Z) via maps f_X, f_Y so that:

Similarity of metric spaces

Two metric spaces (X, d_X) and (Y, d_Y) are similar in the Gromov-Hausdorff Sense if we can map both into a common space (Z, d_Z) via maps f_X, f_Y so that:

1. $(f_X(X), d_Z)$ is isometric to (X, d_X) and $(f_Y(Y), d_Z)$ is isometric to (Y, d_Y).

Similarity of metric spaces

Two metric spaces (X, d_X) and (Y, d_Y) are similar in the Gromov-Hausdorff Sense if we can map both into a common space (Z, d_Z) via maps f_X, f_Y so that:

1. $(f_X(X), d_Z)$ is isometric to (X, d_X) and $(f_Y(Y), d_Z)$ is isometric to (Y, d_Y).
2. The Hausdorff distance between $f_X(X)$ and $f_Y(Y)$ is small.

Similarity of measured metric spaces

We say two *measured* metric spaces $\mathrm{X} = (X, d_X, \mu_X)$ and $\mathrm{Y} = (Y, d_Y, \mu_Y)$ are similar if we can map both into a common space (Z, d_Z) via maps f_X, f_Y so that:

Similarity of measured metric spaces

We say two *measured* metric spaces $\mathrm{X} = (X, d_X, \mu_X)$ and $\mathrm{Y} = (Y, d_Y, \mu_Y)$ are similar if we can map both into a common space (Z, d_Z) via maps f_X, f_Y so that:

1. $(f_X(X), d_Z)$ is isometric to (X, d_X) and $(f_Y(Y), d_Z)$ is isometric to (Y, d_Y).

Similarity of measured metric spaces

We say two *measured* metric spaces $\mathrm{X} = (X, d_X, \mu_X)$ and $\mathrm{Y} = (Y, d_Y, \mu_Y)$ are similar if we can map both into a common space (Z, d_Z) via maps f_X, f_Y so that:

1. $(f_X(X), d_Z)$ is isometric to (X, d_X) and $(f_Y(Y), d_Z)$ is isometric to (Y, d_Y).
2. The Hausdorff distance between $f_X(X)$ and $f_Y(Y)$ is small.

Similarity of measured metric spaces

We say two *measured* metric spaces $\mathrm{X} = (X, d_X, \mu_X)$ and $\mathrm{Y} = (Y, d_Y, \mu_Y)$ are similar if we can map both into a common space (Z, d_Z) via maps f_X, f_Y so that:

1. $(f_X(X), d_Z)$ is isometric to (X, d_X) and $(f_Y(Y), d_Z)$ is isometric to (Y, d_Y).
2. The Hausdorff distance between $f_X(X)$ and $f_Y(Y)$ is small.
3. Prokhorov distance between $\mu_X \circ f_X^{-1}$ and $\mu_Y \circ f_Y^{-1}$ is small.

Similarity of measured metric spaces

We say two *measured* metric spaces $\mathrm{X} = (X, d_X, \mu_X)$ and $\mathrm{Y} = (Y, d_Y, \mu_Y)$ are similar if we can map both into a common space (Z, d_Z) via maps f_X, f_Y so that:

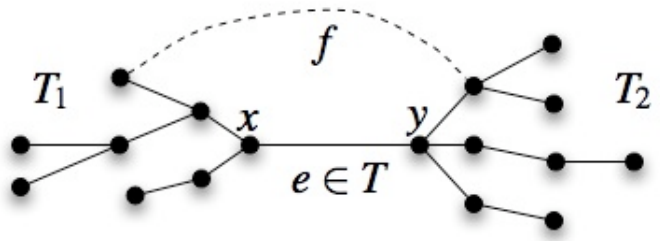
1. $(f_X(X), d_Z)$ is isometric to (X, d_X) and $(f_Y(Y), d_Z)$ is isometric to (Y, d_Y).
2. The Hausdorff distance between $f_X(X)$ and $f_Y(Y)$ is small.
3. Prokhorov distance between $\mu_X \circ f_X^{-1}$ and $\mu_Y \circ f_Y^{-1}$ is small.

This yields *Gromov–Hausdorff–Prokhorov* convergence.

The Kruskal-graph process coupling

For finite G, an edge $e = xy$ is in the MST $\Leftrightarrow \nexists$ a path from x to y all of whose edges have weights less than w_e

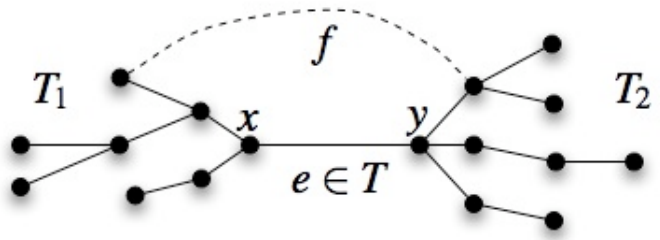
$\forall f$ from T_1 to T_2, $w_f > w_e$.



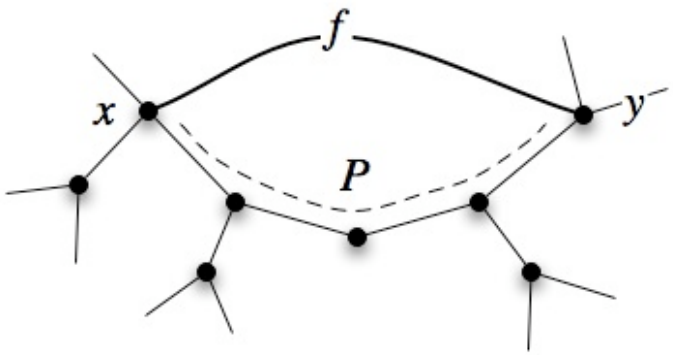
The Kruskal-graph process coupling

For finite G, an edge $e = xy$ is in the MST $\Leftrightarrow \nexists$ a path from x to y all of whose edges have weights less than w_e

$\forall f$ from T_1 to T_2, $w_f > w_e$.



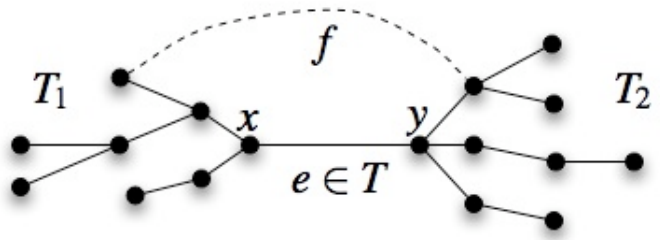
Every edge e of P satisfies $w_e < w_f$



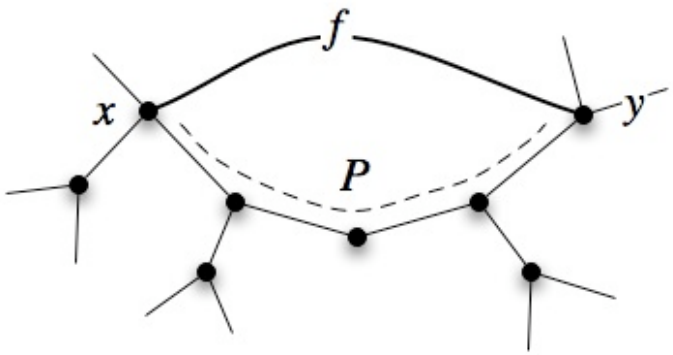
The Kruskal-graph process coupling

For finite G, an edge $e = xy$ is in the MST $\Leftrightarrow \nexists$ a path from x to y all of whose edges have weights less than w_e

$\forall f$ from T_1 to T_2, $w_f > w_e$.



Every edge e of P satisfies $w_e < w_f$

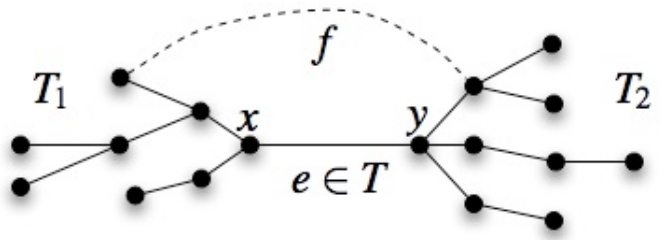


Kruskal's algorithm adds an edge $e = x, y$ of weight p if and only if x and y are in distinct components of $G_{n,p}$.

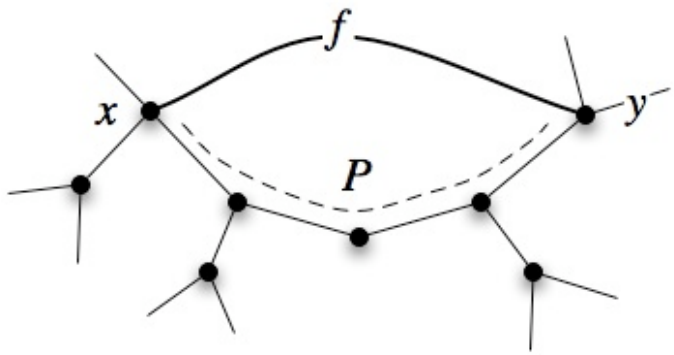
The Kruskal-graph process coupling

For finite G, an edge $e = xy$ is in the MST $\Leftrightarrow \nexists$ a path from x to y all of whose edges have weights less than w_e

$\forall f$ from T_1 to T_2, $w_f > w_e$.



Every edge e of P satisfies $w_e < w_f$



Kruskal's algorithm adds an edge $e = x, y$ of weight p if and only if x and y are in distinct components of $G_{n,p}$.

For all weights p, $G_{n,p}$ has the same set of components as $T_{n,p}$.

Erdős & Renyi, 1960

- For $p = (1 - \epsilon)/n$, a.a.s. every component of $G_{n,p}$ has size $O(\log n)$.

Erdős & Renyi, 1960

- For $p = (1 - \epsilon)/n$, a.a.s. every component of $G_{n,p}$ has size $O(\log n)$.
- For $p = (1 + \epsilon)/n$, a.a.s. one component of $G_{n,p}$ has size $\Theta(n)$, all others have size $O(\log n)$.

Erdős & Renyi, 1960

- For $p = (1 - \epsilon)/n$, a.a.s. every component of $G_{n,p}$ has size $O(\log n)$.
- For $p = (1 + \epsilon)/n$, a.a.s. one component of $G_{n,p}$ has size $\Theta(n)$, all others have size $O(\log n)$.
- For $p = 1/n$, a.a.s. the largest component of $G_{n,p}$ has size $\Theta(n^{2/3})$ and there may be many components of this order.

Erdős & Renyi, 1960

- For $p = (1 - \epsilon)/n$, a.a.s. every component of $G_{n,p}$ has size $O(\log n)$.
- For $p = (1 + \epsilon)/n$, a.a.s. one component of $G_{n,p}$ has size $\Theta(n)$, all others have size $O(\log n)$.
- For $p = 1/n$, a.a.s. the largest component of $G_{n,p}$ has size $\Theta(n^{2/3})$ and there may be many components of this order.

This means that by time $(1 + \epsilon)/n$ the global structure of the minimum spanning tree has essentially been built (effect of remaining connections is to "fill in mass", but doesn't change the geometry).

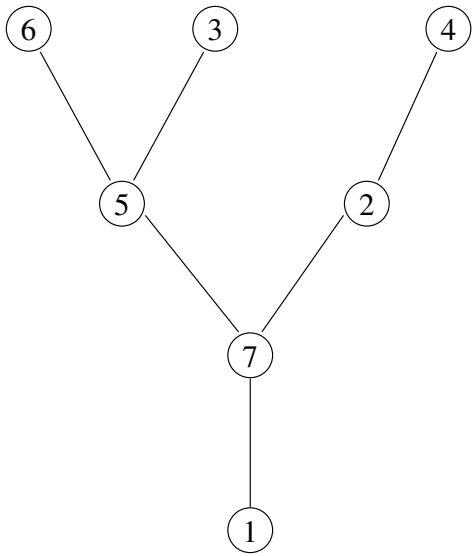
Erdős & Renyi, 1960

- For $p = (1 - \epsilon)/n$, a.a.s. every component of $G_{n,p}$ has size $O(\log n)$.
- For $p = (1 + \epsilon)/n$, a.a.s. one component of $G_{n,p}$ has size $\Theta(n)$, all others have size $O(\log n)$.
- For $p = 1/n$, a.a.s. the largest component of $G_{n,p}$ has size $\Theta(n^{2/3})$ and there may be many components of this order.

This means that by time $(1 + \epsilon)/n$ the global structure of the minimum spanning tree has essentially been built (effect of remaining connections is to "fill in mass", but doesn't change the geometry).
So to understand the tree, we really need to understand the graph process at time around $1/n$.

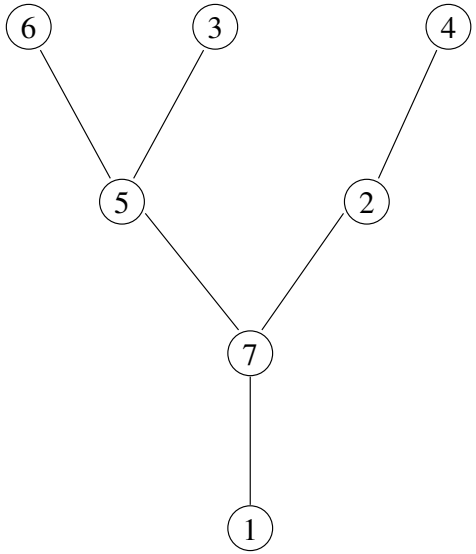
A reminder about trees.

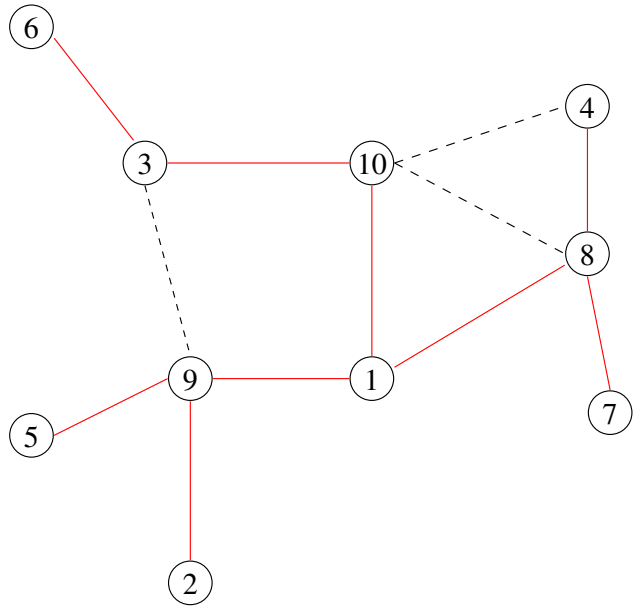
A tree is a connected graph with no cycles. A tree always has one more vertex than it has edges.



A reminder about trees.

A tree is a connected graph with no cycles. A tree always has one more vertex than it has edges.





A connected graph which is not a tree has surplus equal to the number of edges more than a tree that it has. The graph on the right has surplus 3.

Convergence of component sizes of $G_{n,1/n}$.

- Let $C_1^{(n)}, C_2^{(n)}, \ldots$ be the sizes of the connected components of $G_{n,1/n}$, listed in decreasing order, let $S_1^{(n)}, S_2^{(n)}, \ldots$ be their surplusses.

Convergence of component sizes of $G_{n,1/n}$.

- Let $C_1^{(n)}, C_2^{(n)}, \ldots$ be the sizes of the connected components of $G_{n,1/n}$, listed in decreasing order, let $S_1^{(n)}, S_2^{(n)}, \ldots$ be their surplusses.

Theorem (Aldous, 1997)

$$(C_1^{(n)}/n^{2/3}, C_2^{(n)}/n^{2/3}, \ldots) \to (L_1, L_2, \ldots), \quad \textit{and}$$
$$(S_1^{(n)}, S_2^{(n)}, \ldots) \to (S_1, S_2, \ldots),$$

jointly in distribution, as $n \to \infty$.

Convergence of component sizes of $G_{n,1/n}$.

- Let $C_1^{(n)}, C_2^{(n)}, \ldots$ be the sizes of the connected components of $G_{n,1/n}$, listed in decreasing order, let $S_1^{(n)}, S_2^{(n)}, \ldots$ be their surplusses.

Theorem (Aldous, 1997)

$$(C_1^{(n)}/n^{2/3}, C_2^{(n)}/n^{2/3}, \ldots) \to (L_1, L_2, \ldots), \quad \textit{and}$$
$$(S_1^{(n)}, S_2^{(n)}, \ldots) \to (S_1, S_2, \ldots),$$

jointly in distribution, as $n \to \infty$.

- The first convergence is in the space of sequences $\mathbf{x} = (x_1, x_2, \ldots)$ with $\sum x_i^2 < \infty$ and with distance

$$d(\mathbf{x}, \mathbf{y}) = \sqrt{\sum (x_i - y_i)^2}.$$

Convergence of component sizes of $G_{n,1/n}$.

- Let $C_1^{(n)}, C_2^{(n)}, \ldots$ be the sizes of the connected components of $G_{n,1/n}$, listed in decreasing order, let $S_1^{(n)}, S_2^{(n)}, \ldots$ be their surplusses.

Theorem (Aldous, 1997)

$$(C_1^{(n)}/n^{2/3}, C_2^{(n)}/n^{2/3}, \ldots) \to (L_1, L_2, \ldots), \quad \textit{and}$$
$$(S_1^{(n)}, S_2^{(n)}, \ldots) \to (S_1, S_2, \ldots),$$

jointly in distribution, as $n \to \infty$.

- The first convergence is in the space of sequences $\mathbf{x} = (x_1, x_2, \ldots)$ with $\sum x_i^2 < \infty$ and with distance

$$d(\mathbf{x}, \mathbf{y}) = \sqrt{\sum (x_i - y_i)^2}.$$

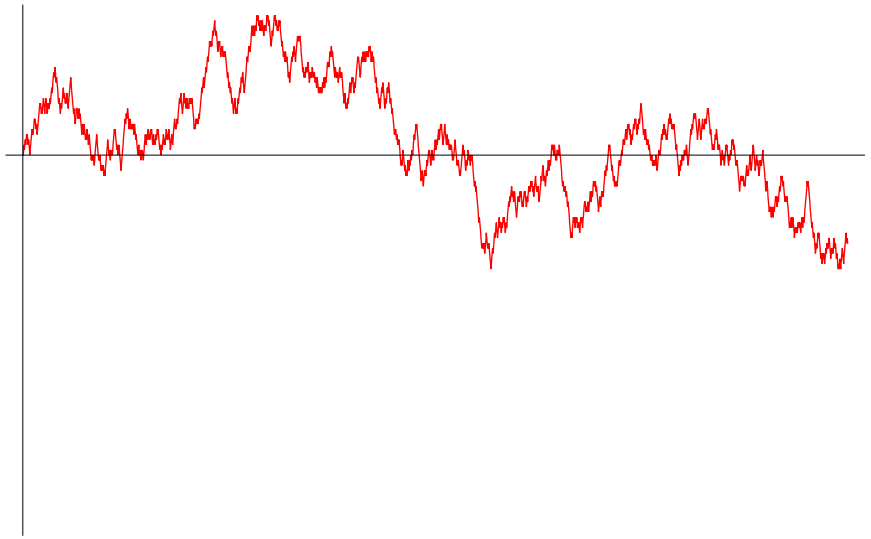
- The second is of finite-dimensional distributions.

Convergence of component sizes of $G_{n,1/n}$.

What are the limit sequences?

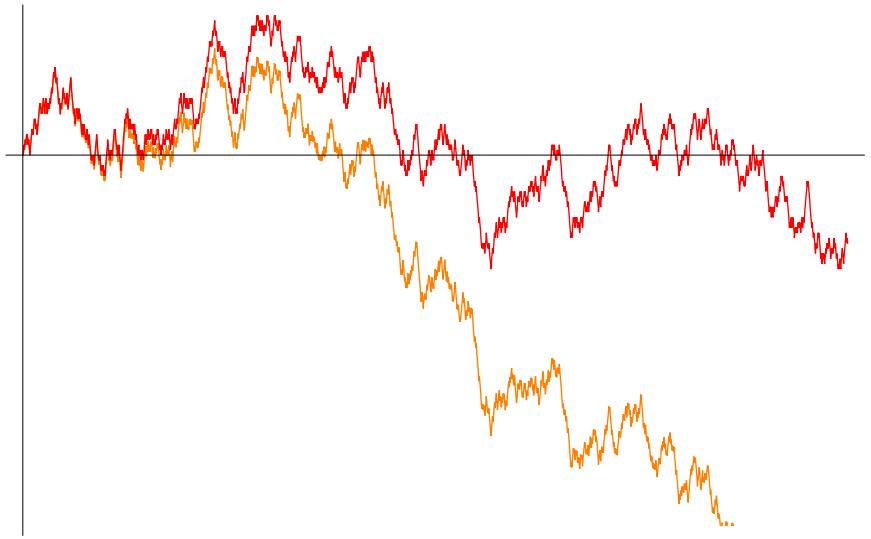
Convergence of component sizes of $G_{n,1/n}$.

- Let $\{B_t\}_{t\geq 0}$ be a standard Brownian motion



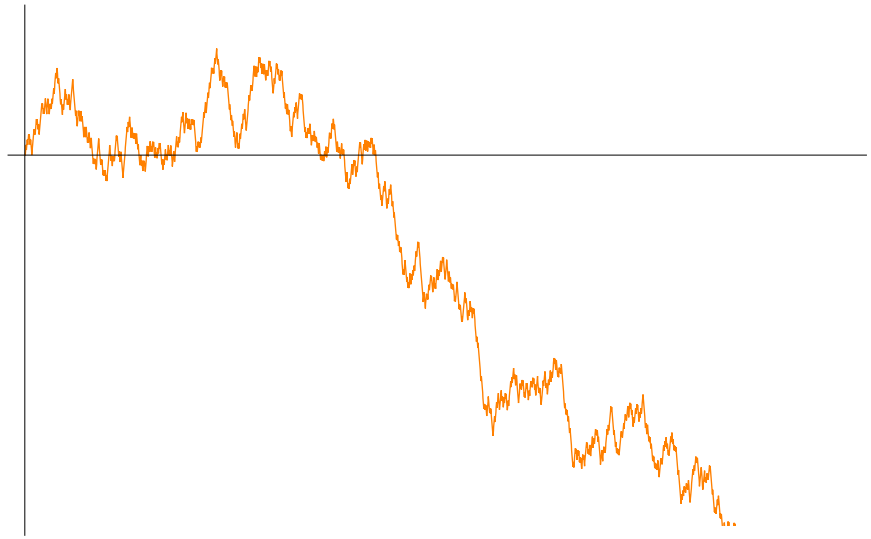
Convergence of component sizes of $G_{n,1/n}$.

- Let $\{B_t\}_{t\geq 0}$ be a standard Brownian motion
- Let $W_t = B_t - t^2/2$.



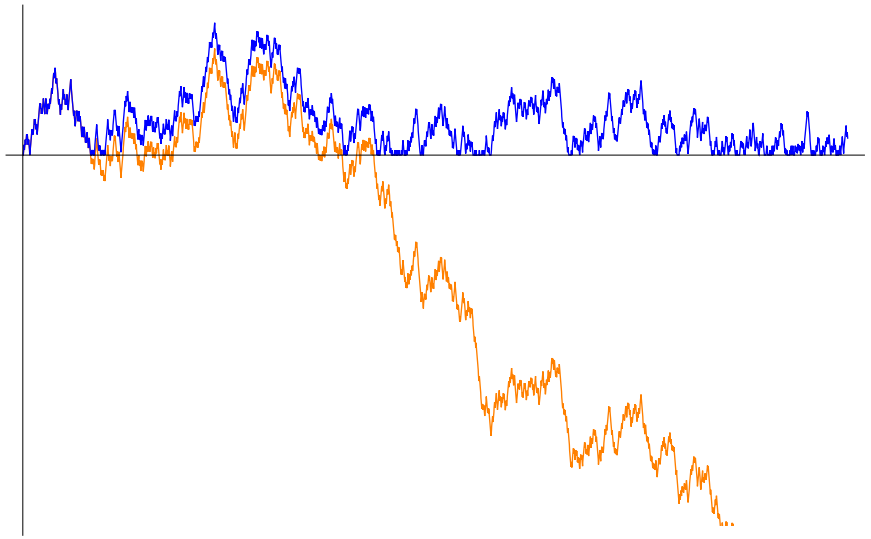
Convergence of component sizes of $G_{n,1/n}$.

- Let $\{B_t\}_{t\geq 0}$ be a standard Brownian motion
- Let $W_t = B_t - t^2/2$.



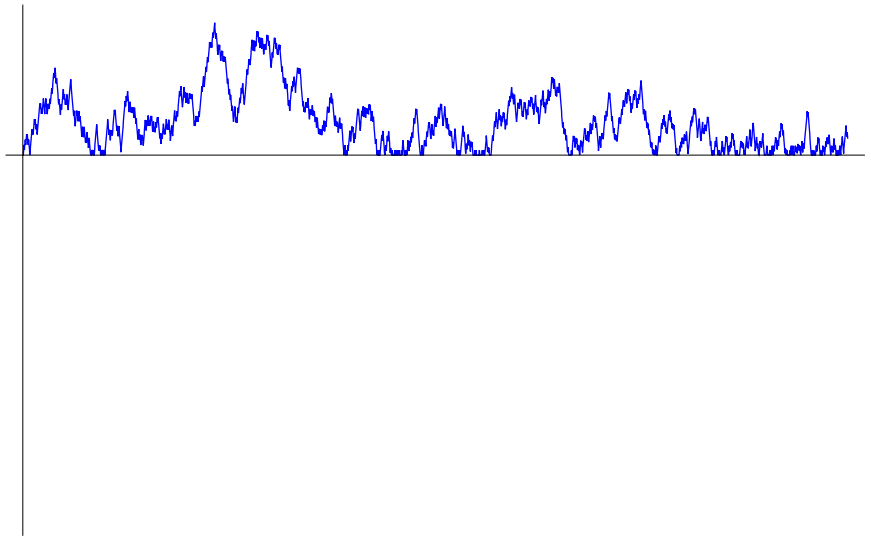
Convergence of component sizes of $G_{n,1/n}$.

- Let $\{B_t\}_{t\geq 0}$ be a standard Brownian motion
- Let $W_t = B_t - t^2/2$.
- Let $W_t^* = W_t - \inf_{0\leq s\leq t} W_s$.



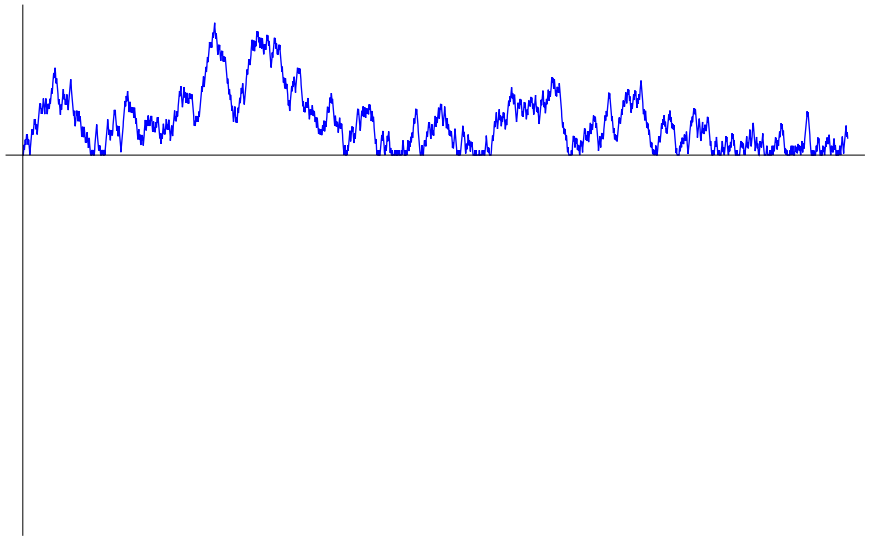
Convergence of component sizes of $G_{n,1/n}$.

- Let $\{B_t\}_{t\geq 0}$ be a standard Brownian motion
- Let $W_t = B_t - t^2/2$.
- Let $W_t^* = W_t - \inf_{0\leq s\leq t} W_s$.



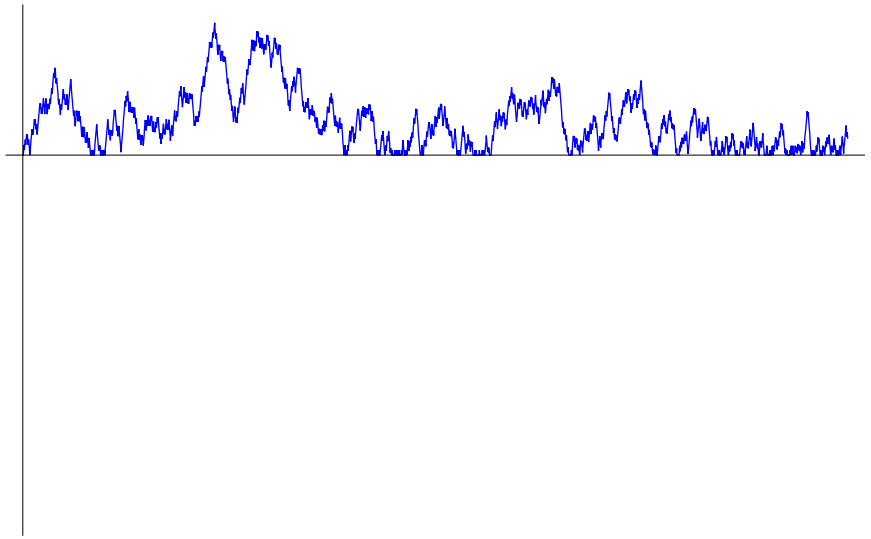
Convergence of component sizes of $G_{n,1/n}$.

- The first limit object $(L_1, L_2, \ldots)$ is the sequence of the lengths of the excursions above 0 of W_t^* (in decreasing order).



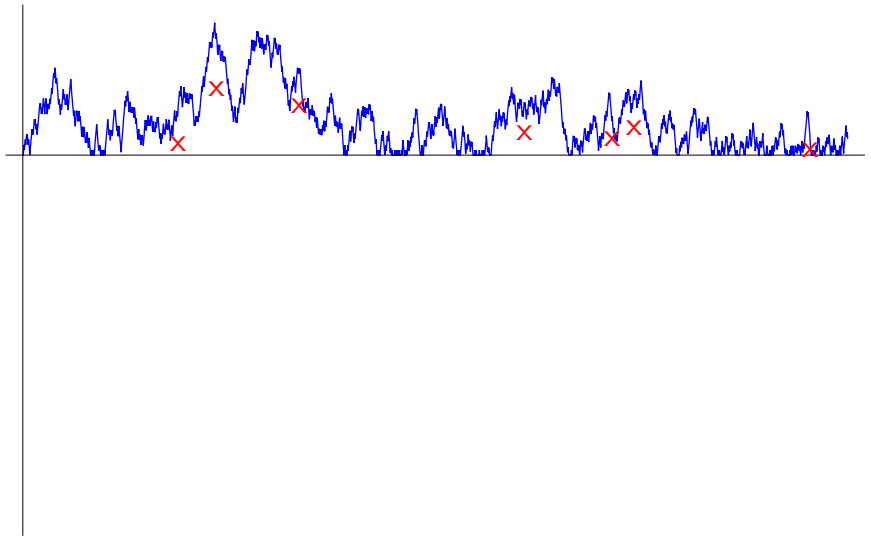
Convergence of component sizes of $G_{n,1/n}$.

- The first limit object $(L_1, L_2, \ldots)$ is the sequence of the lengths of the excursions above 0 of W_t^* (in decreasing order).
- Let $\mathcal{P}$ be a homogeneous Poisson process of "marks" under the excursions of W_t^*.



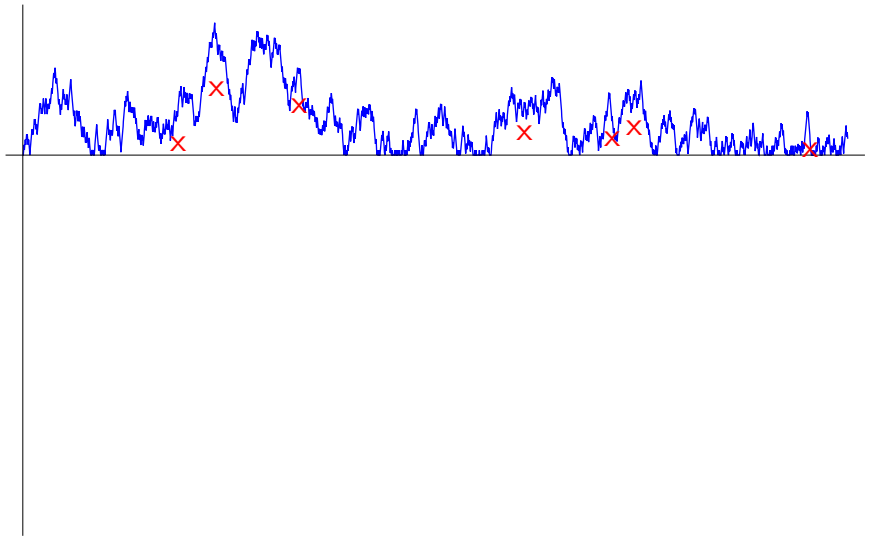
Convergence of component sizes of $G_{n,1/n}$.

- The first limit object $(L_1, L_2, \ldots)$ is the sequence of the lengths of the excursions above 0 of W_t^* (in decreasing order).
- Let $\mathcal{P}$ be a homogeneous Poisson process of "marks" under the excursions of W_t^*.



Convergence of component sizes of $G_{n,1/n}$.

- The first limit object $(L_1, L_2, \ldots)$ is the sequence of the lengths of the excursions above 0 of W_t^* (in decreasing order).
- Let $\mathcal{P}$ be a homogeneous Poisson process of "marks" under the excursions of W_t^*.



- Then S_i is distributed as the number of marks under the excursion with length L_i.

What? Why?

Why should "component size and surplus" be encoded by "excursion plus marks"?

What? Why?

Why should "component size and surplus" be encoded by "excursion plus marks"?

The right way to think of it:

- The *excursion* is encoding a certain spanning tree of the component.

What? Why?

Why should "component size and surplus" be encoded by "excursion plus marks"?

The right way to think of it:

- The *excursion* is encoding a certain spanning tree of the component.
- The *marks* are telling you where to put the extra edges to recover the component from its spanning tree.

What? Why?

Why should "component size and surplus" be encoded by "excursion plus marks"?

The right way to think of it:

- The *excursion* is encoding a certain spanning tree of the component.
- The *marks* are telling you where to put the extra edges to recover the component from its spanning tree.

So I need to explain to you:

What? Why?

Why should "component size and surplus" be encoded by "excursion plus marks"?

The right way to think of it:

- The *excursion* is encoding a certain spanning tree of the component.
- The *marks* are telling you where to put the extra edges to recover the component from its spanning tree.

So I need to explain to you:

- How excursions encode trees;

What? Why?

Why should "component size and surplus" be encoded by "excursion plus marks"?

The right way to think of it:

- The *excursion* is encoding a certain spanning tree of the component.
- The *marks* are telling you where to put the extra edges to recover the component from its spanning tree.

So I need to explain to you:

- How excursions encode trees; and,

What? Why?

Why should "component size and surplus" be encoded by "excursion plus marks"?

The right way to think of it:

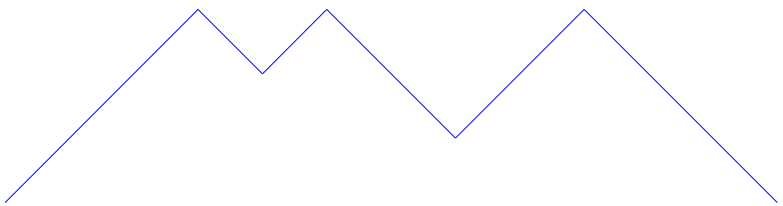
- The *excursion* is encoding a certain spanning tree of the component.
- The *marks* are telling you where to put the extra edges to recover the component from its spanning tree.

So I need to explain to you:

- How excursions encode trees; and,
- How marks tell you where to put extra edges.

Trees coded by continuous functions.

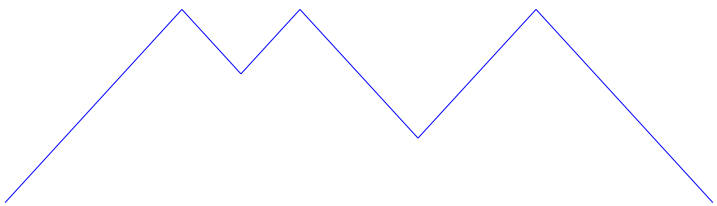
Given any continuous excursion, we can define an associated tree (metric space), by gluing together points of the excursion that (a) have the same height and (b) can see each other.



[Picture by Christina Goldschmidt.]

Trees coded by continuous functions.

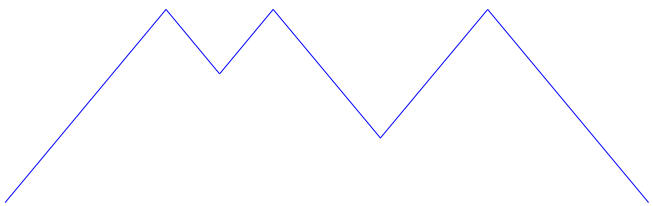
Given any continuous excursion, we can define an associated tree (metric space), by gluing together points of the excursion that (a) have the same height and (b) can see each other.



[Picture by Christina Goldschmidt.]

Trees coded by continuous functions.

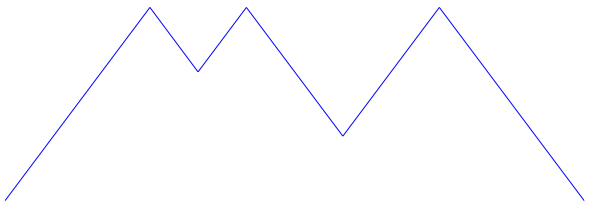
Given any continuous excursion, we can define an associated tree (metric space), by gluing together points of the excursion that (a) have the same height and (b) can see each other.



[Picture by Christina Goldschmidt.]

Trees coded by continuous functions.

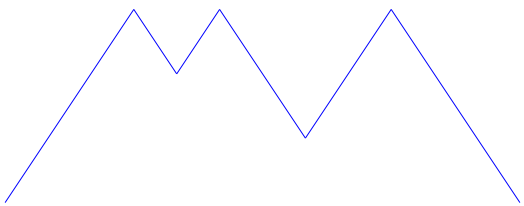
Given any continuous excursion, we can define an associated tree (metric space), by gluing together points of the excursion that (a) have the same height and (b) can see each other.



[Picture by Christina Goldschmidt.]

Trees coded by continuous functions.

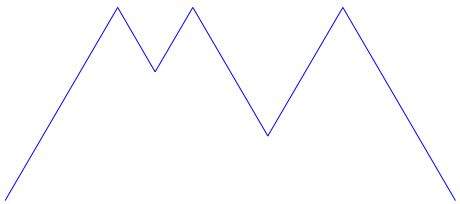
Given any continuous excursion, we can define an associated tree (metric space), by gluing together points of the excursion that (a) have the same height and (b) can see each other.



[Picture by Christina Goldschmidt.]

Trees coded by continuous functions.

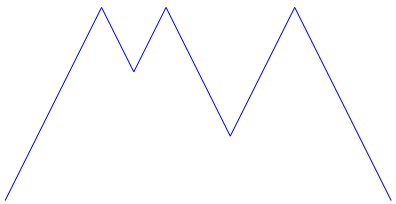
Given any continuous excursion, we can define an associated tree (metric space), by gluing together points of the excursion that (a) have the same height and (b) can see each other.



[Picture by Christina Goldschmidt.]

Trees coded by continuous functions.

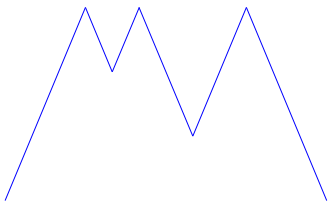
Given any continuous excursion, we can define an associated tree (metric space), by gluing together points of the excursion that (a) have the same height and (b) can see each other.



[Picture by Christina Goldschmidt.]

Trees coded by continuous functions.

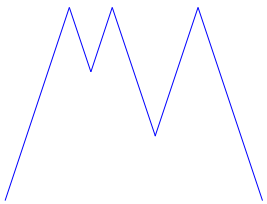
Given any continuous excursion, we can define an associated tree (metric space), by gluing together points of the excursion that (a) have the same height and (b) can see each other.



[Picture by Christina Goldschmidt.]

Trees coded by continuous functions.

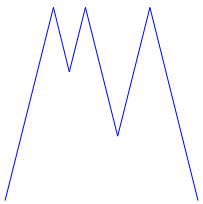
Given any continuous excursion, we can define an associated tree (metric space), by gluing together points of the excursion that (a) have the same height and (b) can see each other.



[Picture by Christina Goldschmidt.]

Trees coded by continuous functions.

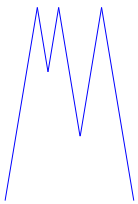
Given any continuous excursion, we can define an associated tree (metric space), by gluing together points of the excursion that (a) have the same height and (b) can see each other.



[Picture by Christina Goldschmidt.]

Trees coded by continuous functions.

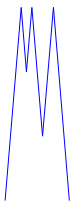
Given any continuous excursion, we can define an associated tree (metric space), by gluing together points of the excursion that (a) have the same height and (b) can see each other.



[Picture by Christina Goldschmidt.]

Trees coded by continuous functions.

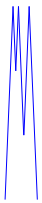
Given any continuous excursion, we can define an associated tree (metric space), by gluing together points of the excursion that (a) have the same height and (b) can see each other.



[Picture by Christina Goldschmidt.]

Trees coded by continuous functions.

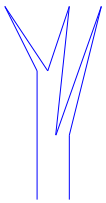
Given any continuous excursion, we can define an associated tree (metric space), by gluing together points of the excursion that (a) have the same height and (b) can see each other.



[Picture by Christina Goldschmidt.]

Trees coded by continuous functions.

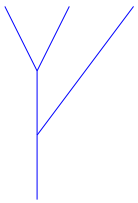
Given any continuous excursion, we can define an associated tree (metric space), by gluing together points of the excursion that (a) have the same height and (b) can see each other.



[Picture by Christina Goldschmidt.]

Trees coded by continuous functions.

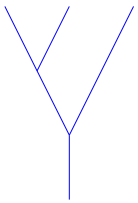
Given any continuous excursion, we can define an associated tree (metric space), by gluing together points of the excursion that (a) have the same height and (b) can see each other.



[Picture by Christina Goldschmidt.]

Trees coded by continuous functions.

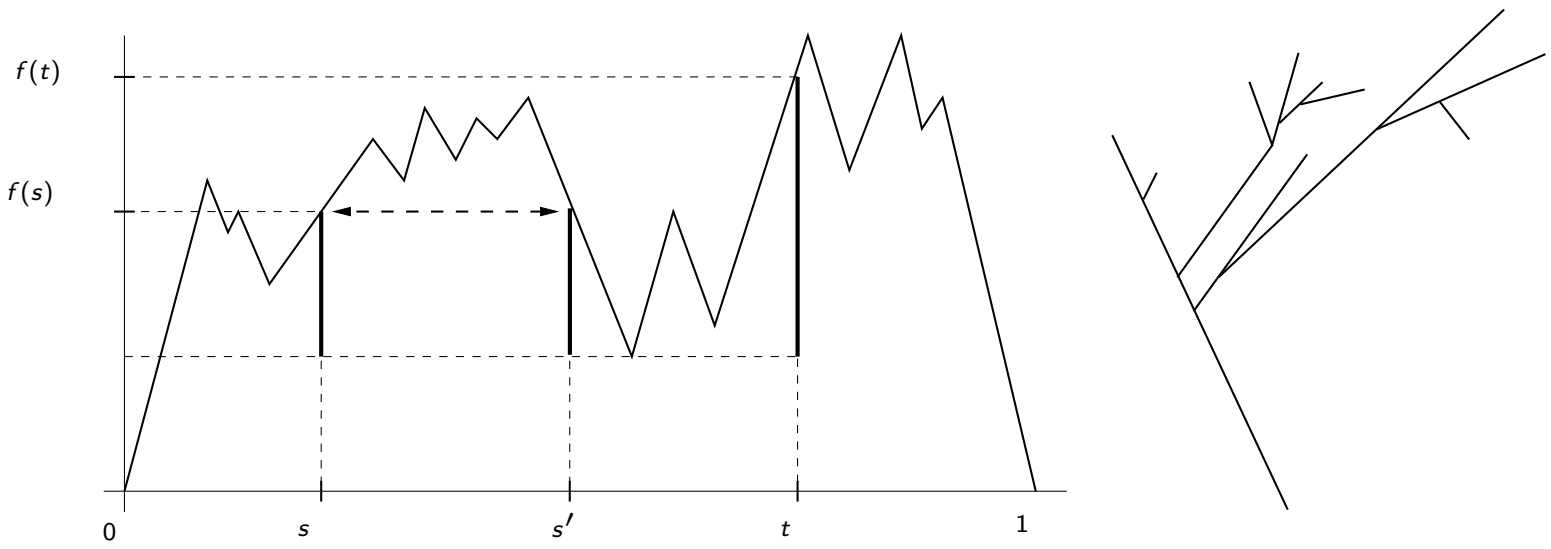
Given any continuous excursion, we can define an associated tree (metric space), by gluing together points of the excursion that (a) have the same height and (b) can see each other.



[Picture by Christina Goldschmidt.]

Trees coded by continuous functions.

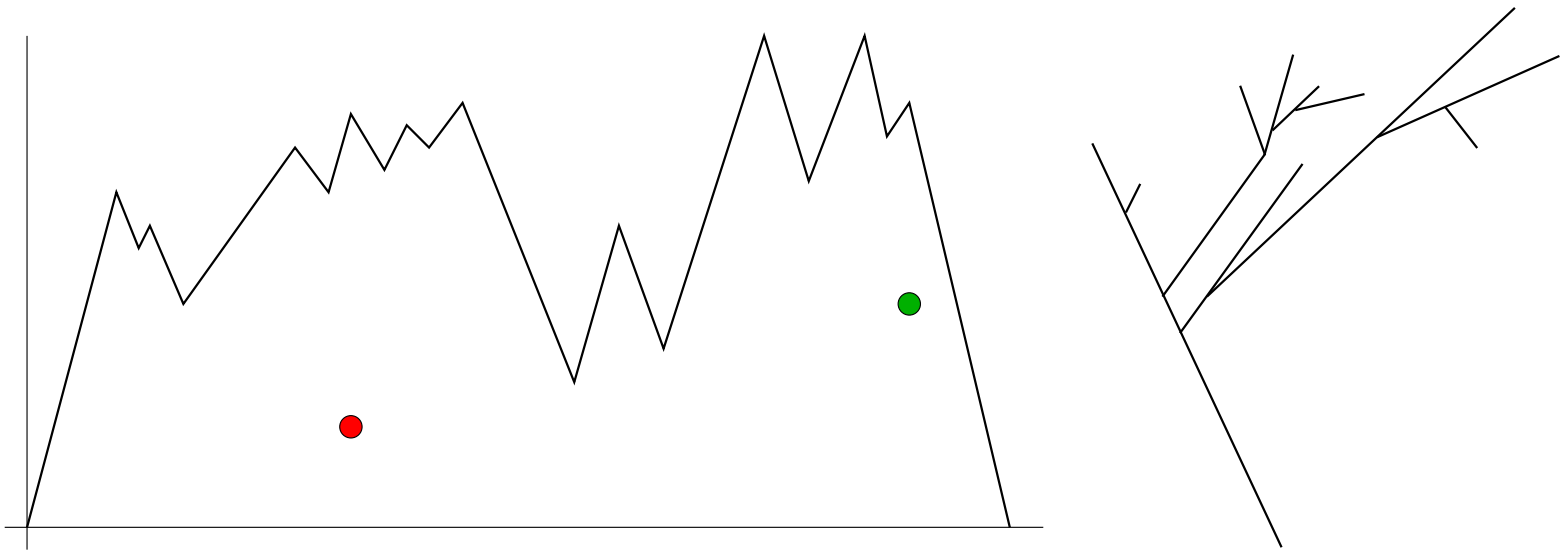
Given any continuous excursion, we can define an associated tree (metric space), by gluing together points of the excursion that (a) have the same height and (b) can see each other.



[Picture by Marie Albenque.]

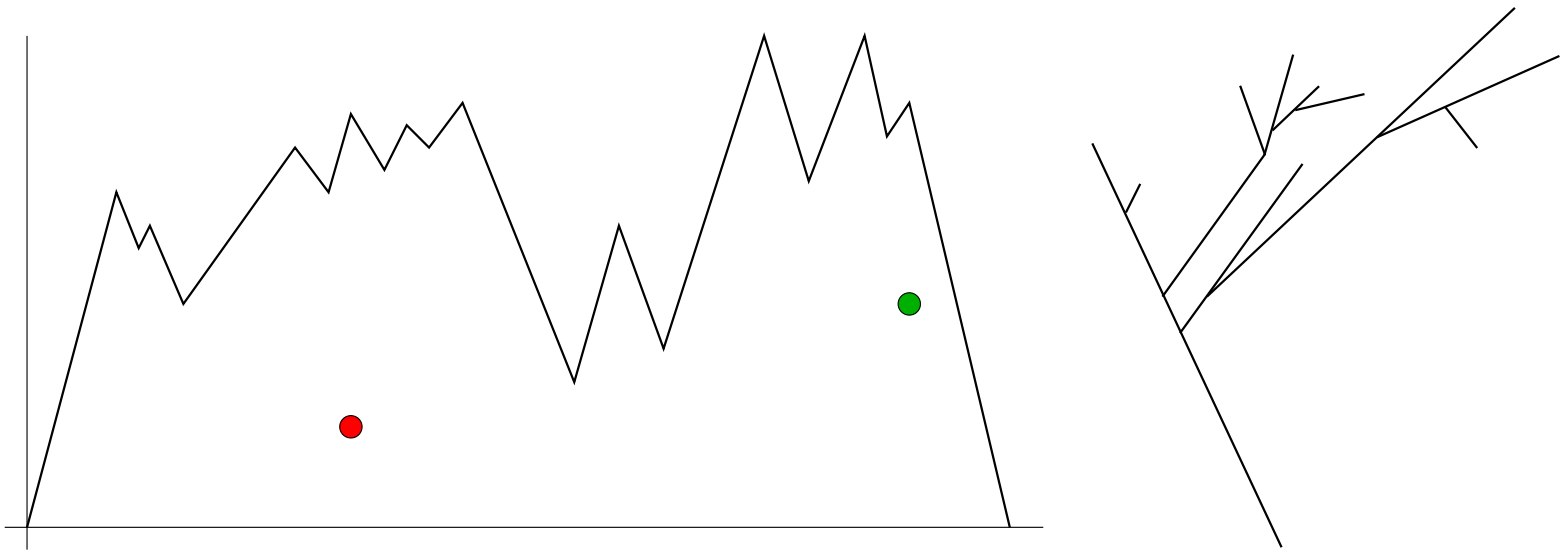
Handling the surplus edges.

We saw how to associate a tree to a continuous excursion by gluing together points of the excursion that (a) have the same height and (b) can see each other.



Handling the surplus edges.

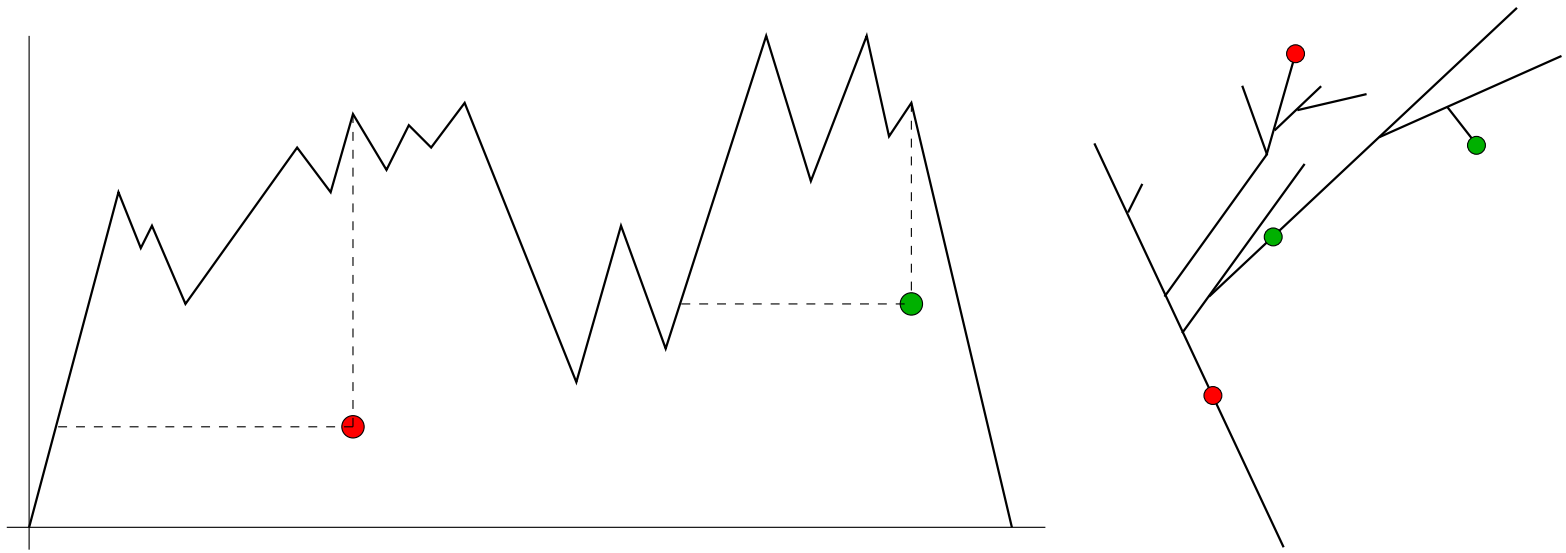
We saw how to associate a tree to a continuous excursion by gluing together points of the excursion that (a) have the same height and (b) can see each other.



A *mark* under the excursion naturally identifies two points of the associated tree.

Handling the surplus edges.

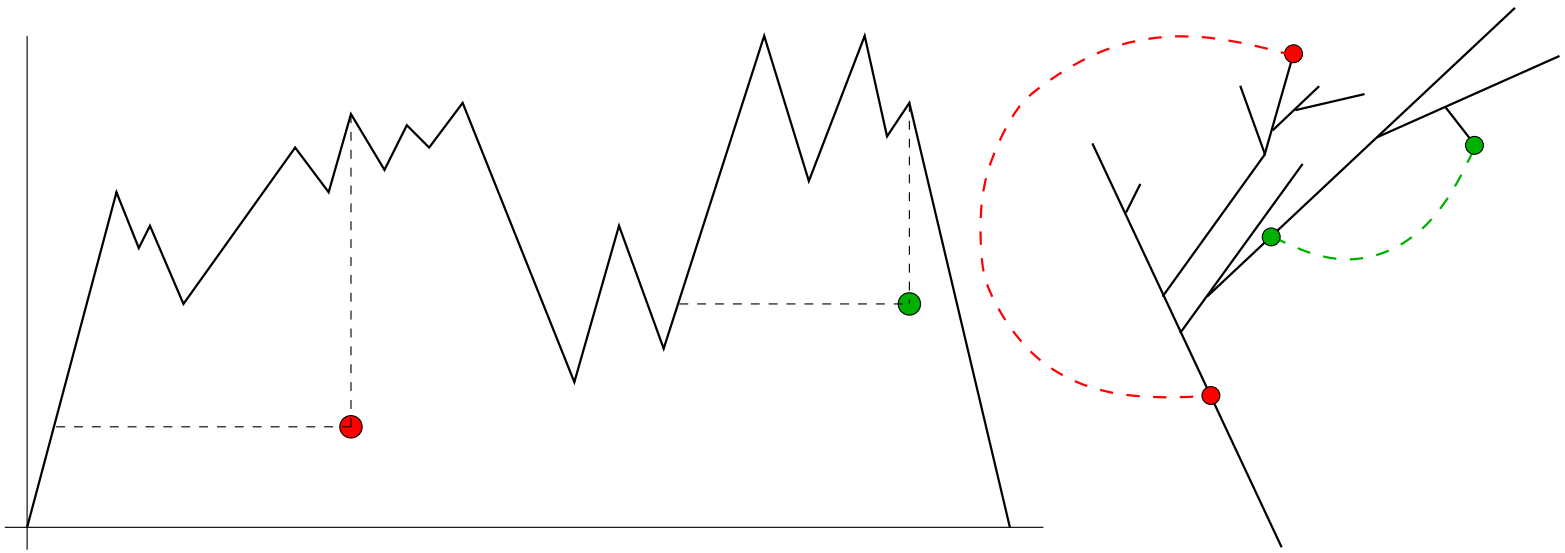
We saw how to associate a tree to a continuous excursion by gluing together points of the excursion that (a) have the same height and (b) can see each other.



A *mark* under the excursion naturally identifies two points of the associated tree.

Handling the surplus edges.

We saw how to associate a tree to a continuous excursion by gluing together points of the excursion that (a) have the same height and (b) can see each other.



A *mark* under the excursion naturally identifies two points of the associated tree.

We obtain a new metric space by identifying these points.

What kind of random trees?.

- Let $\mathcal{C}$ be a component of $G_{n,1/n}$ with m vertices, and suppose we condition $\mathcal{C}$ to be a tree.

What kind of random trees?.

- Let $\mathcal{C}$ be a component of $G_{n,1/n}$ with m vertices, and suppose we condition $\mathcal{C}$ to be a tree.
- Then $\mathcal{C}$ is distributed as a *uniform random* tree on m labelled vertices; any spanning tree of the vertex set is equally likely. (There are m^{m-2} such possible trees.)

Uniform random trees.

So fix an integer m and generate a uniform random tree on labels $1, 2, \ldots, m$.

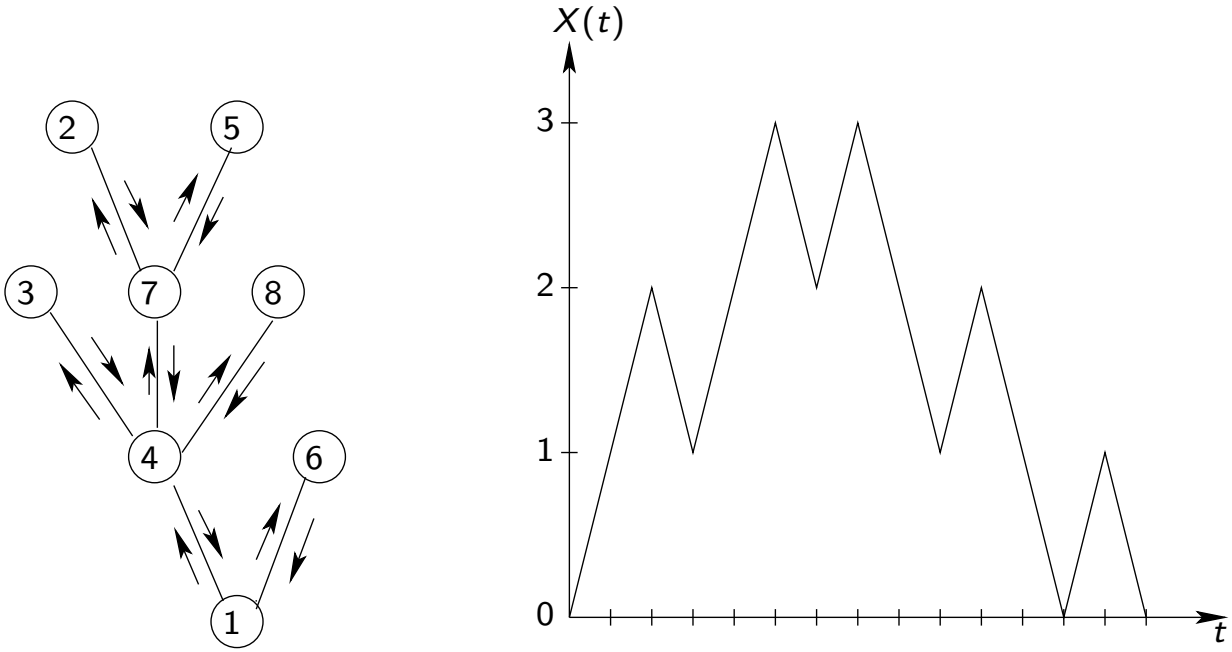
Uniform random trees.

So fix an integer m and generate a uniform random tree on labels $1, 2, \ldots, m$.

These trees are well understood; they are distributed as Poisson Galton-Watson trees conditional on their size. In particular, they have height $\Theta(\sqrt{m})$.

The Harris walk.

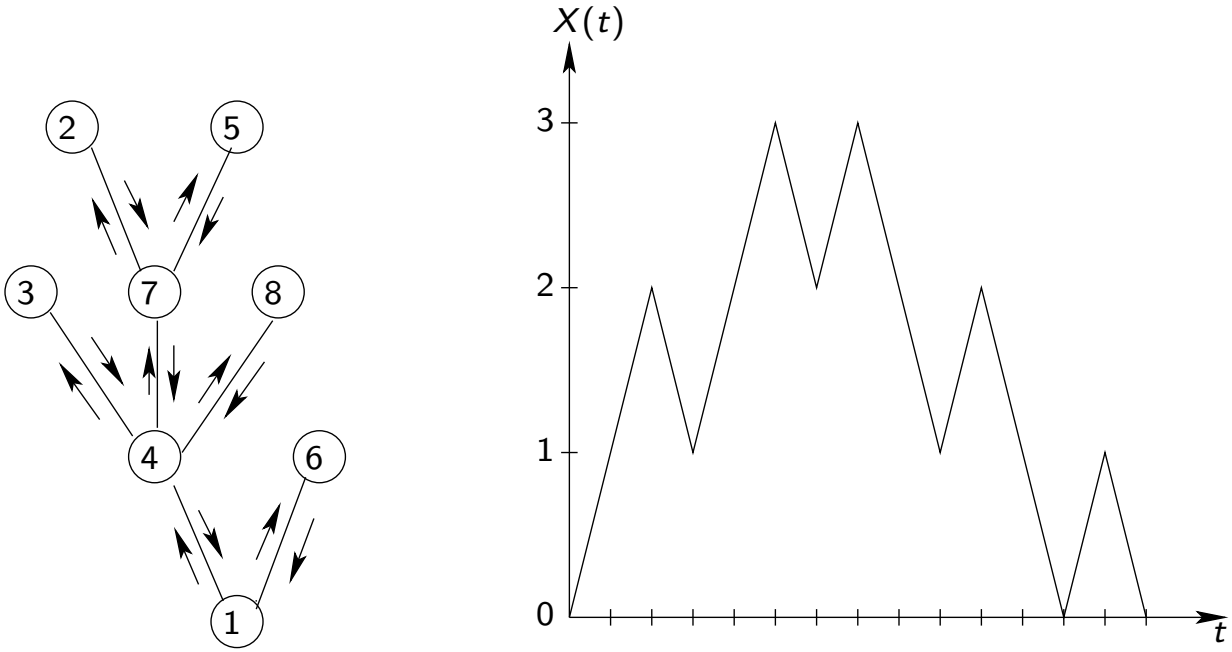
The Harris walk is an excursion encoding heights in a finite tree.



[Picture by Marie Albenque.]

The Harris walk.

The Harris walk is an excursion encoding heights in a finite tree.



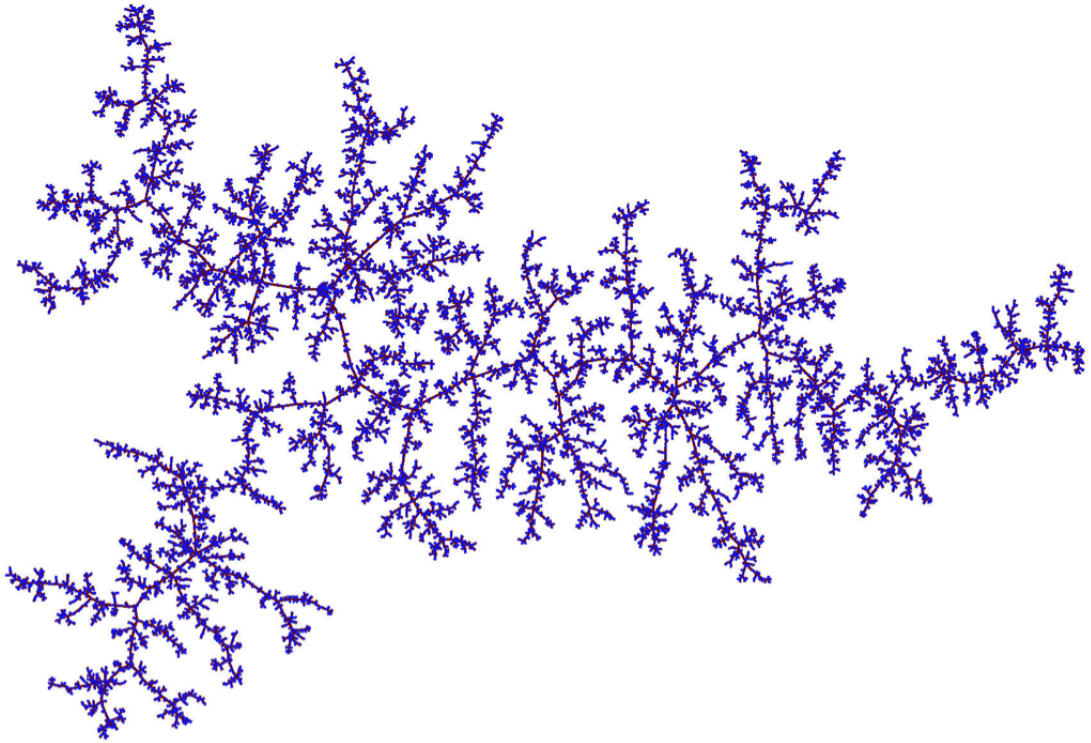
[Picture by Marie Albenque.]

When the tree is uniformly random on $1, 2, \ldots, m$, this essentially looks like a random walk with $2m$ steps, conditioned to stay positive and return to zero at time $2m$.

Rescaled, its limit is a Brownian excursion.

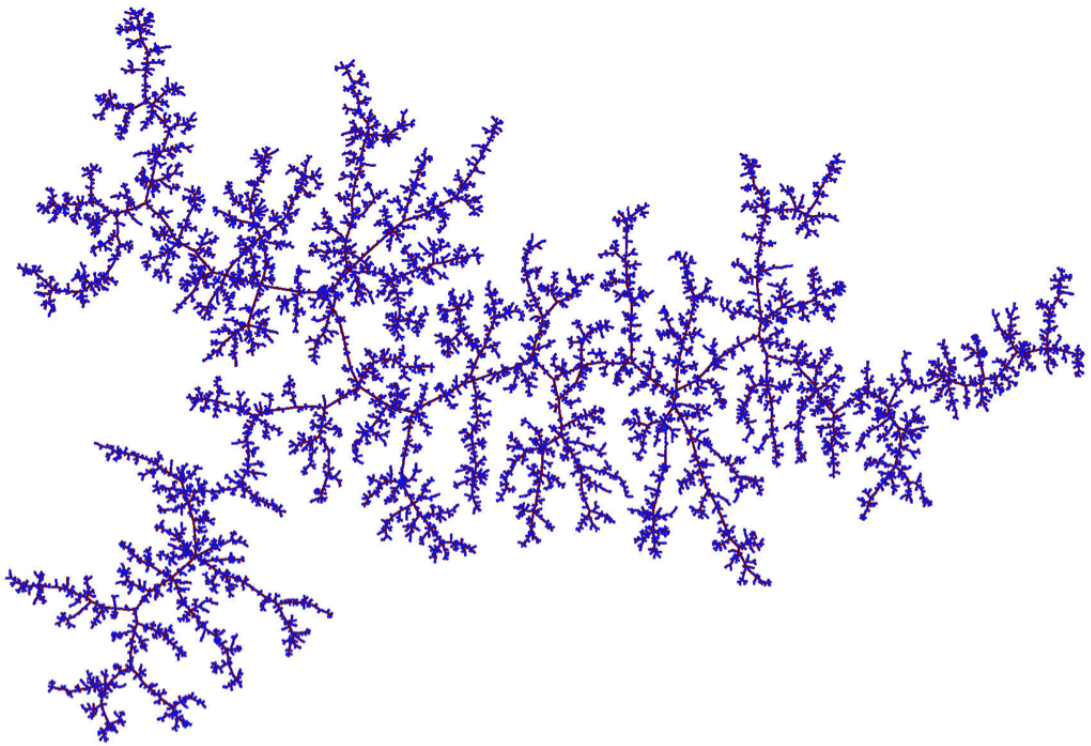
The continuum random tree.

If the *tree itself* is rescaled (edge lengths $1/\sqrt{m}$ instead of 1), then as $m \to \infty$, there is a limiting object, called the Brownian continuum random tree.



The continuum random tree.

If the *tree itself* is rescaled (edge lengths $1/\sqrt{m}$ instead of 1), then as $m \to \infty$, there is a limiting object, called the Brownian continuum random tree.



This is just the tree coded by a standard Brownian excursion.

Metric convergence of uniform trees.

The uniform random tree on $1, \ldots, m$, with distances rescaled by $1/\sqrt{m}$, and mass $1/m$ for each vertex, converges in distribution to the Brownian continuum random tree, in the Gromov-Hausdorff-Prokhorov sense.

Limiting random graph: what kind of limit.

Let $\mathcal{C}^{(n)} = (\mathcal{C}_1^{(n)}, \mathcal{C}_2^{(n)}, \ldots)$ be the size-ordered sequence of components of $G_{n,1/n}$, with distances rescaled by $n^{-1/3}$ and masses rescaled by $n^{-2/3}$.

Limiting random graph: what kind of limit.

Let $\mathcal{C}^{(n)} = (\mathcal{C}_1^{(n)}, \mathcal{C}_2^{(n)}, \ldots)$ be the size-ordered sequence of components of $G_{n,1/n}$, with distances rescaled by $n^{-1/3}$ and masses rescaled by $n^{-2/3}$.
We proved that

$$(\mathcal{C}_1^{(n)}, \mathcal{C}_2^{(n)}, \ldots) \xrightarrow{d} (\mathcal{M}_1, \mathcal{M}_2, \ldots).$$

Limiting random graph: what kind of limit.

Let $\mathcal{C}^{(n)} = (\mathcal{C}_1^{(n)}, \mathcal{C}_2^{(n)}, \ldots)$ be the size-ordered sequence of components of $G_{n,1/n}$, with distances rescaled by $n^{-1/3}$ and masses rescaled by $n^{-2/3}$.
We proved that

$$(\mathcal{C}_1^{(n)}, \mathcal{C}_2^{(n)}, \ldots) \xrightarrow{d} (\mathcal{M}_1, \mathcal{M}_2, \ldots).$$

Here convergence is with respect to the metric

$$d(\mathcal{C}, \mathcal{M}) = \left(\sum_{i=1}^{\infty} d_{GHP}(\mathcal{C}_i, \mathcal{M}_i)^4 \right)^{1/4}.$$

Proof idea.

- Let $\mathcal{C}$ be a component of $G_{n,1/n}$, and suppose we condition $\mathcal{C}$ to have m vertices and s surplus edges.

Proof idea.

- Let $\mathcal{C}$ be a component of $G_{n,1/n}$, and suppose we condition $\mathcal{C}$ to have m vertices and s surplus edges.
- Then $\mathcal{C}$ is distributed as a *uniform random* connected graph on m labelled vertices with s surplus edges (I am equally likely to pick each of connected graphs with surplus s on those labels.)

Proof idea.

- Let $\mathcal{C}$ be a component of $G_{n,1/n}$, and suppose we condition $\mathcal{C}$ to have m vertices and s surplus edges.
- Then $\mathcal{C}$ is distributed as a *uniform random* connected graph on m labelled vertices with s surplus edges (I am equally likely to pick each of connected graphs with surplus s on those labels.)
- We will canonically associate to each such graph a *discrete excursion*, together with a set of marks under the excursion.

Proof idea.

- Let $\mathcal{C}$ be a component of $G_{n,1/n}$, and suppose we condition $\mathcal{C}$ to have m vertices and s surplus edges.
- Then $\mathcal{C}$ is distributed as a *uniform random* connected graph on m labelled vertices with s surplus edges (I am equally likely to pick each of connected graphs with surplus s on those labels.)
- We will canonically associate to each such graph a *discrete excursion*, together with a set of marks under the excursion.
- This excursion corresponds to a canonical choice of spanning tree (in fact, it is just the spanning tree whose edge set is lexicographically minimal,

Proof idea.

- Let $\mathcal{C}$ be a component of $G_{n,1/n}$, and suppose we condition $\mathcal{C}$ to have m vertices and s surplus edges.
- Then $\mathcal{C}$ is distributed as a *uniform random* connected graph on m labelled vertices with s surplus edges (I am equally likely to pick each of connected graphs with surplus s on those labels.)
- We will canonically associate to each such graph a *discrete excursion*, together with a set of marks under the excursion.
- This excursion corresponds to a canonical choice of spanning tree (in fact, it is just the spanning tree whose edge set is lexicographically minimal, when each vertex is are relabelled by the *sequence* of labels on the path from the root).

Proof idea.

- Let $\mathcal{C}$ be a component of $G_{n,1/n}$, and suppose we condition $\mathcal{C}$ to have m vertices and s surplus edges.
- Then $\mathcal{C}$ is distributed as a *uniform random* connected graph on m labelled vertices with s surplus edges (I am equally likely to pick each of connected graphs with surplus s on those labels.)
- We will canonically associate to each such graph a *discrete excursion*, together with a set of marks under the excursion.
- This excursion corresponds to a canonical choice of spanning tree (in fact, it is just the spanning tree whose edge set is lexicographically minimal, when each vertex is are relabelled by the *sequence* of labels on the path from the root).
- Warning: the excursion is *not* the Harris walk of this spanning tree.

Depth-first exploration

The canonical excursion is constructed via a procedure called depth-first exploration.

Depth-first exploration

The canonical excursion is constructed via a procedure called depth-first exploration.

We will explore the graph step-by-step. At each step, vertices will have the possibility to be current, alive or dead.

Depth-first exploration

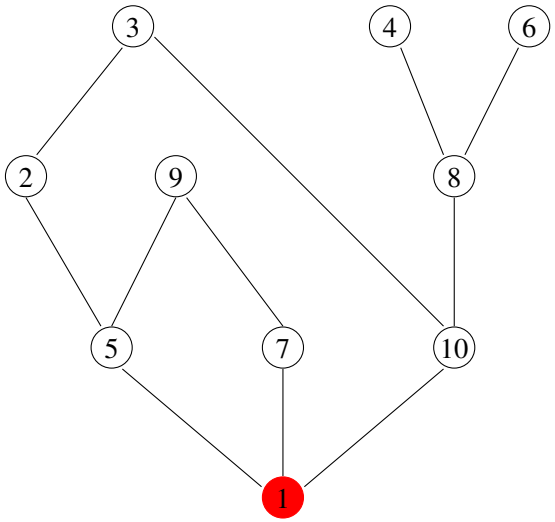
The canonical excursion is constructed via a procedure called depth-first exploration.

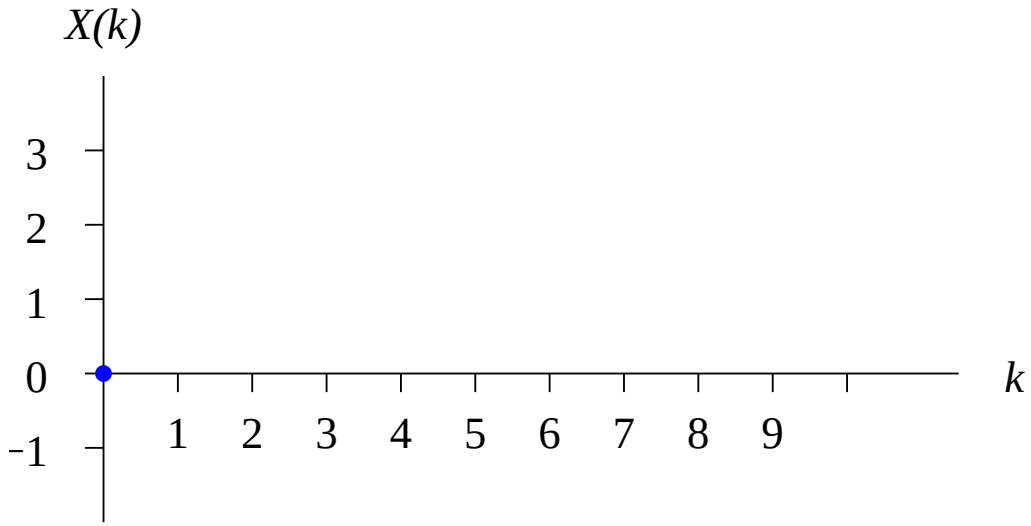
We will explore the graph step-by-step. At each step, vertices will have the possibility to be current, alive or dead.

We will also want to keep track of some information. At step k, let $X(k)$ be the number of vertices which are alive.

Depth-first walk

Step 0: initialization



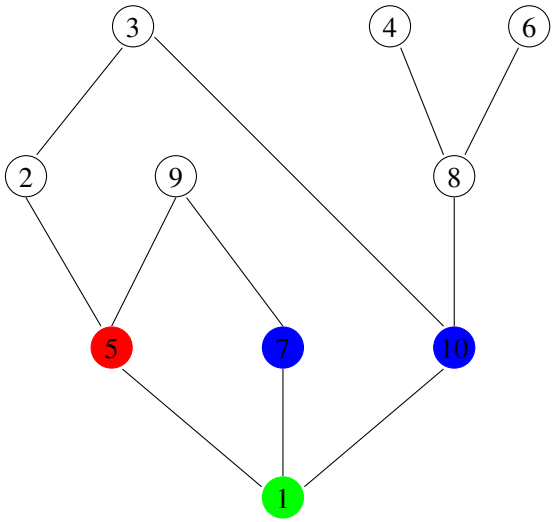


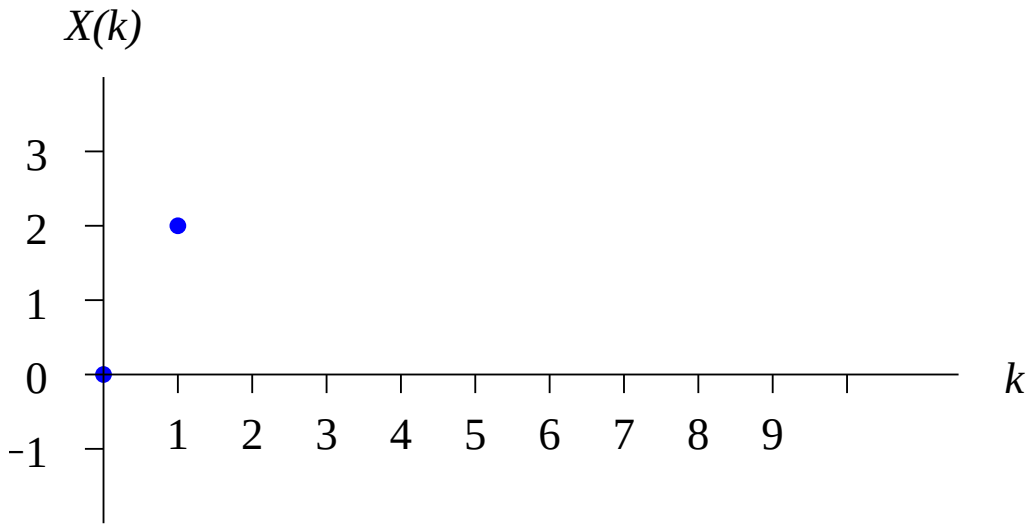
[Picture by Christina Goldschmidt.]

Current: 1 Alive: none Dead: none

Depth-first walk

Step 1



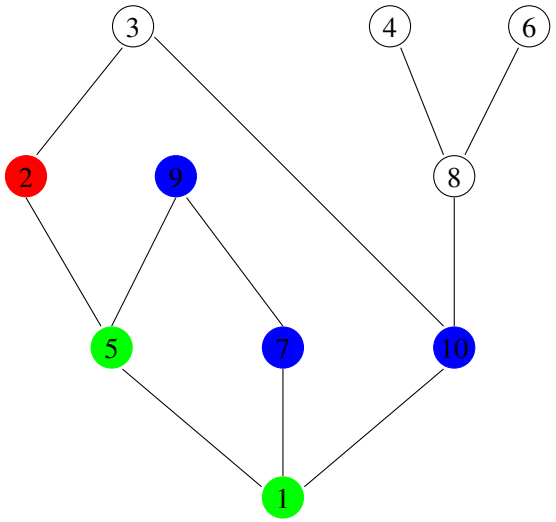


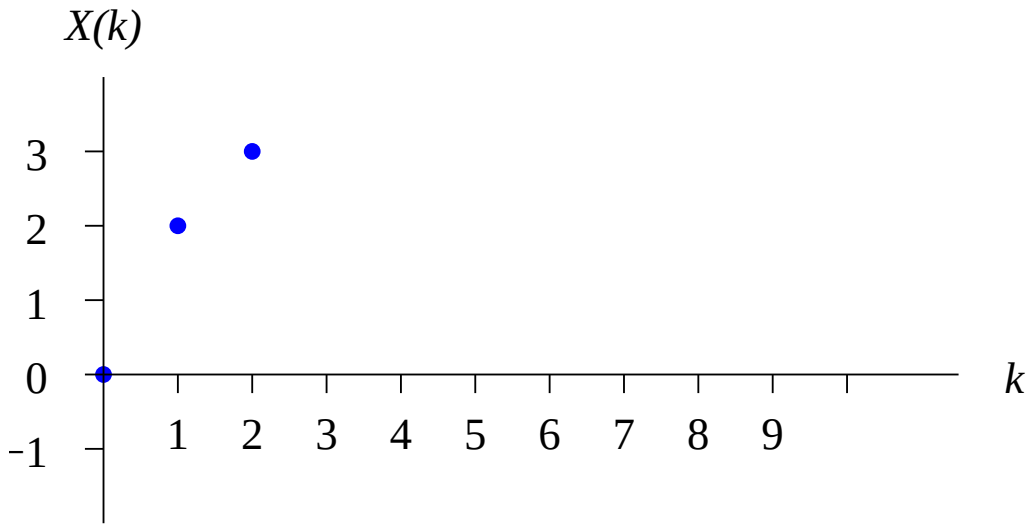
[Picture by Christina Goldschmidt.]

Current: 5 Alive: 7,10 Dead: 1

Depth-first walk

Step 2

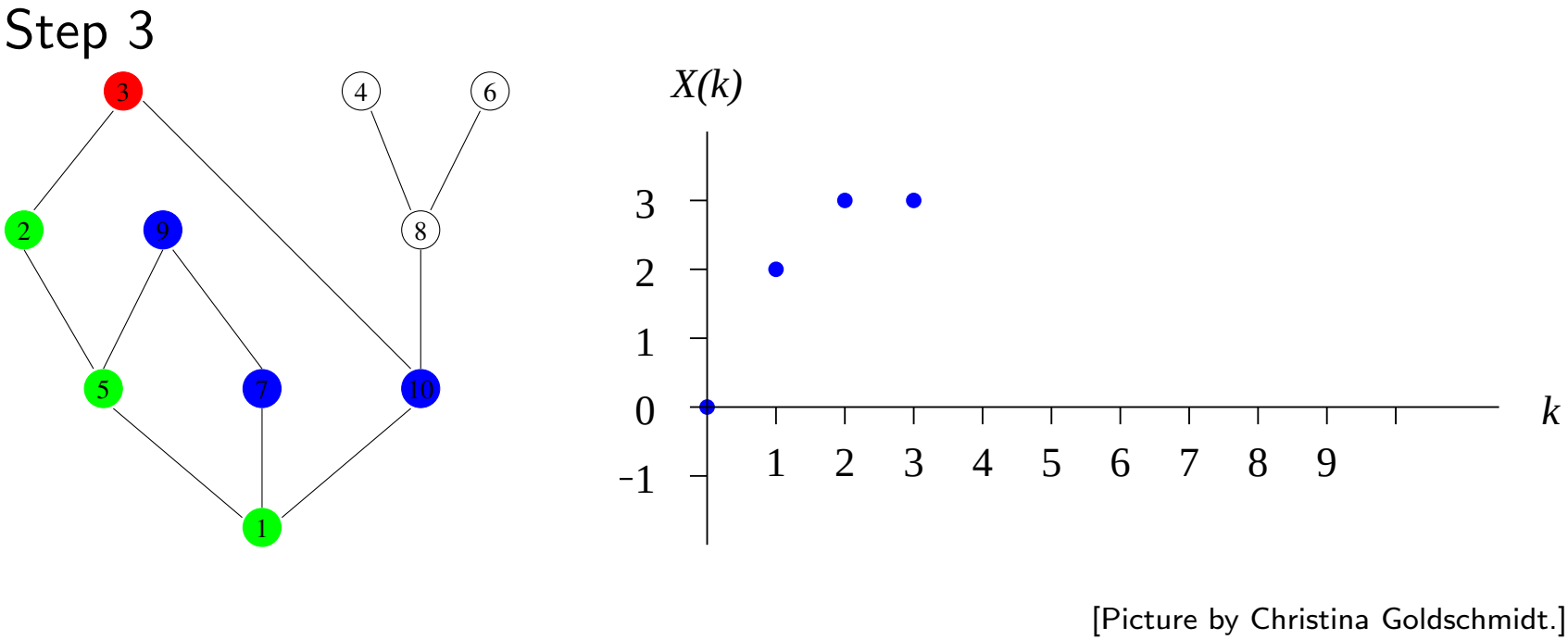




[Picture by Christina Goldschmidt.]

Current: 2 Alive: 9,7,10 Dead: 1,5

Depth-first walk

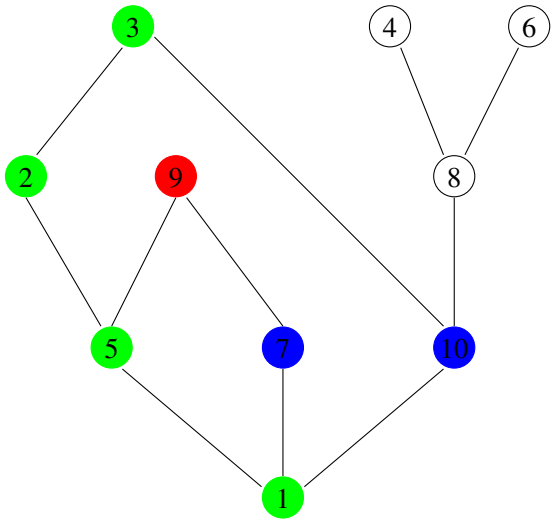


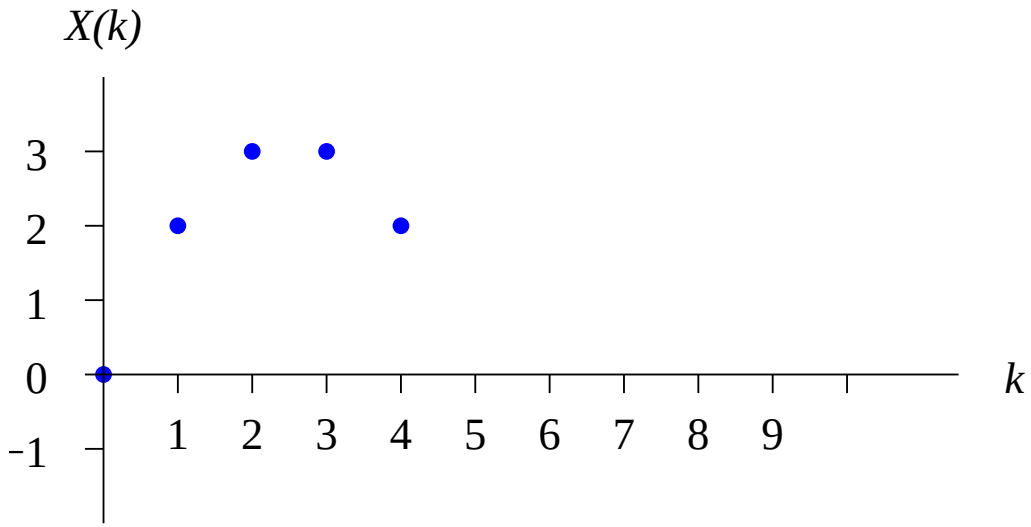
[Picture by Christina Goldschmidt.]

Current: 3 Alive: 9,7,10 Dead: 1,5,2

Depth-first walk

Step 4

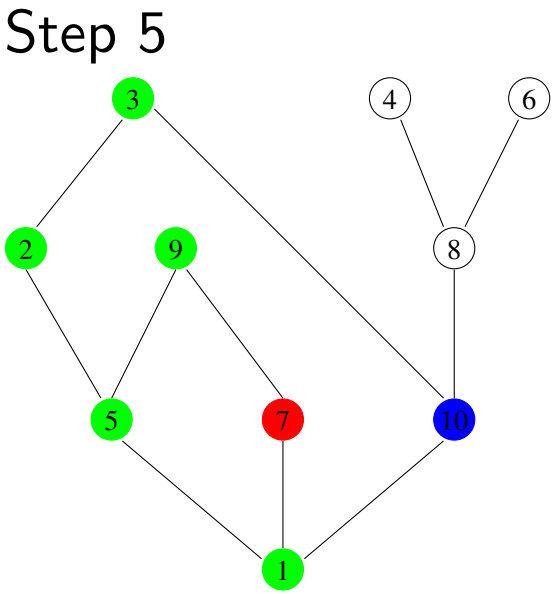


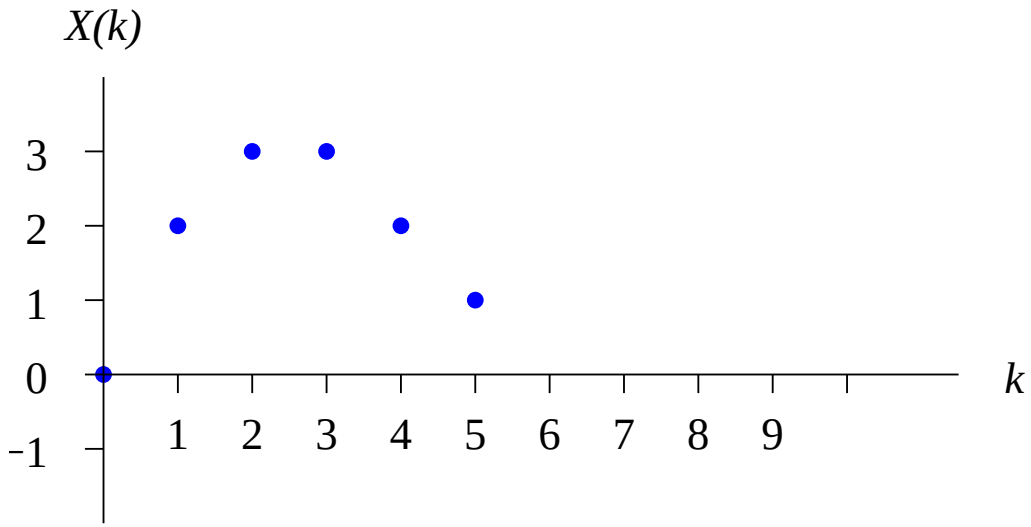


[Picture by Christina Goldschmidt.]

Current: 9 Alive: 7,10 Dead: 1,5,2,3

Depth-first walk

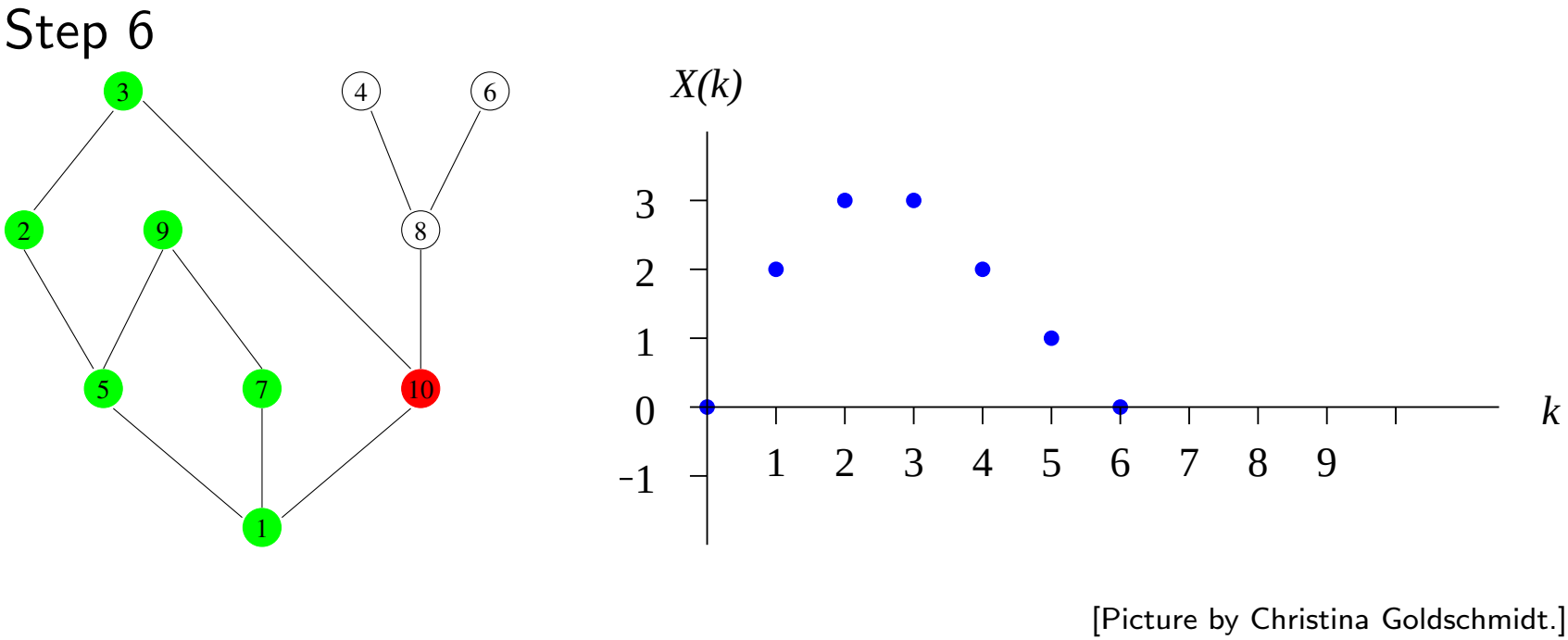




[Picture by Christina Goldschmidt.]

Current: 7 Alive: 10 Dead: 1,5,2,3,9

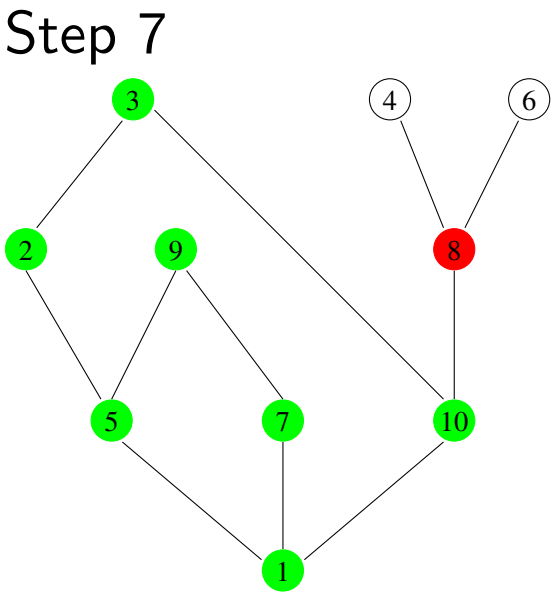
Depth-first walk

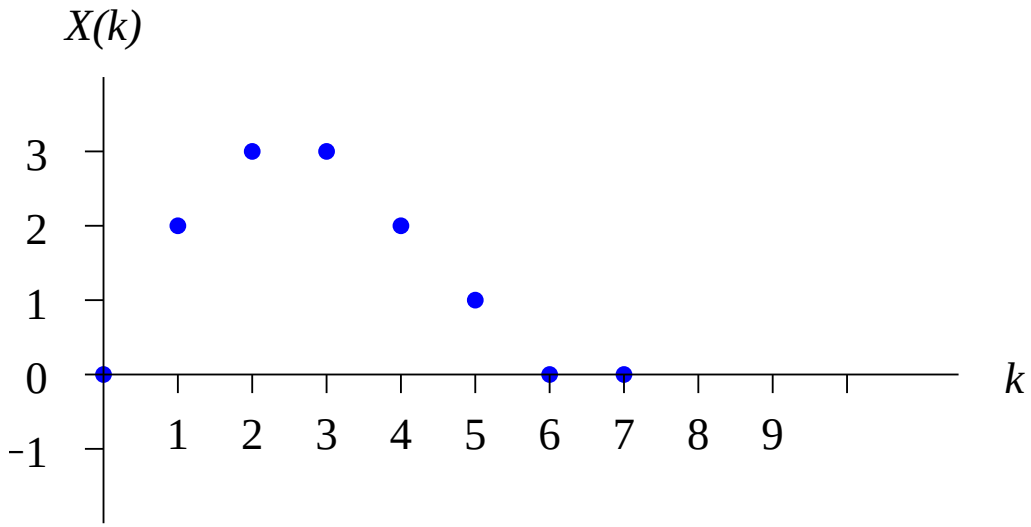


[Picture by Christina Goldschmidt.]

Current: 10 Alive: none Dead: 1,5,2,3,9,7

Depth-first walk

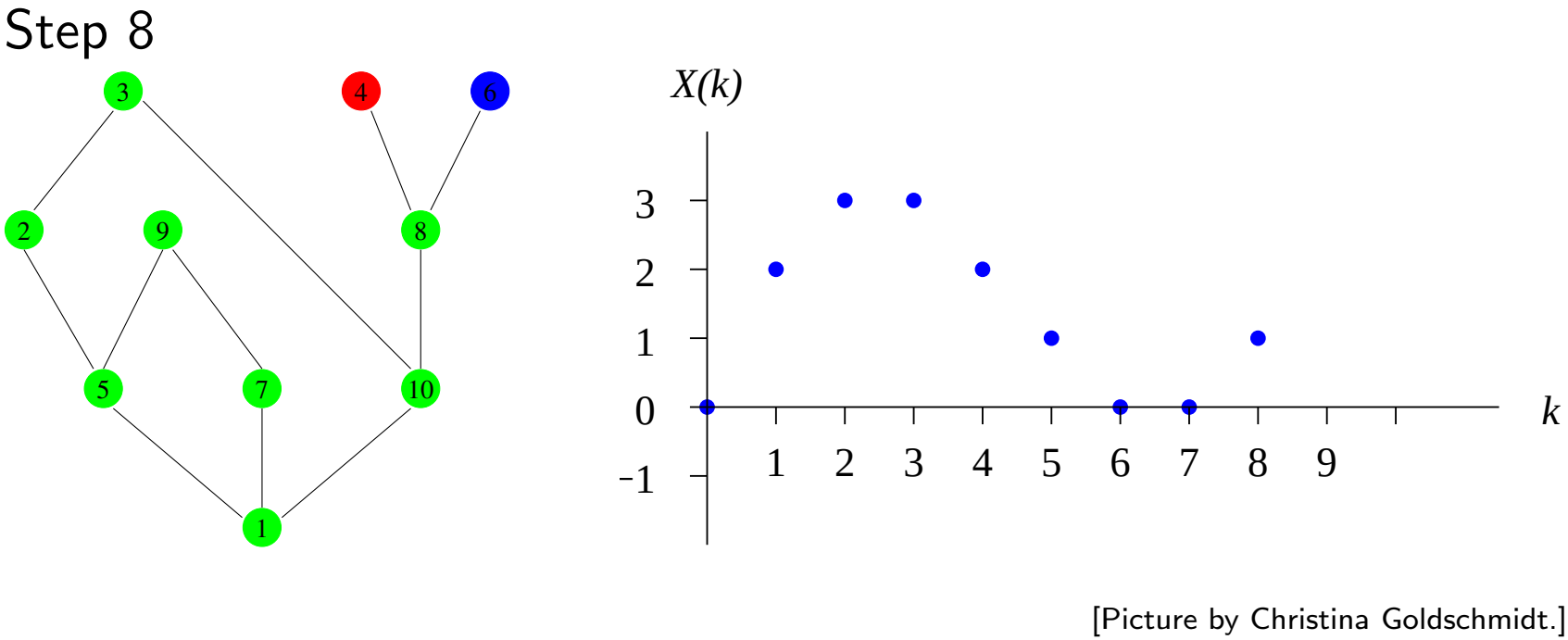




[Picture by Christina Goldschmidt.]

Current: 8 Alive: none Dead: 1,5,2,3,9,7,10

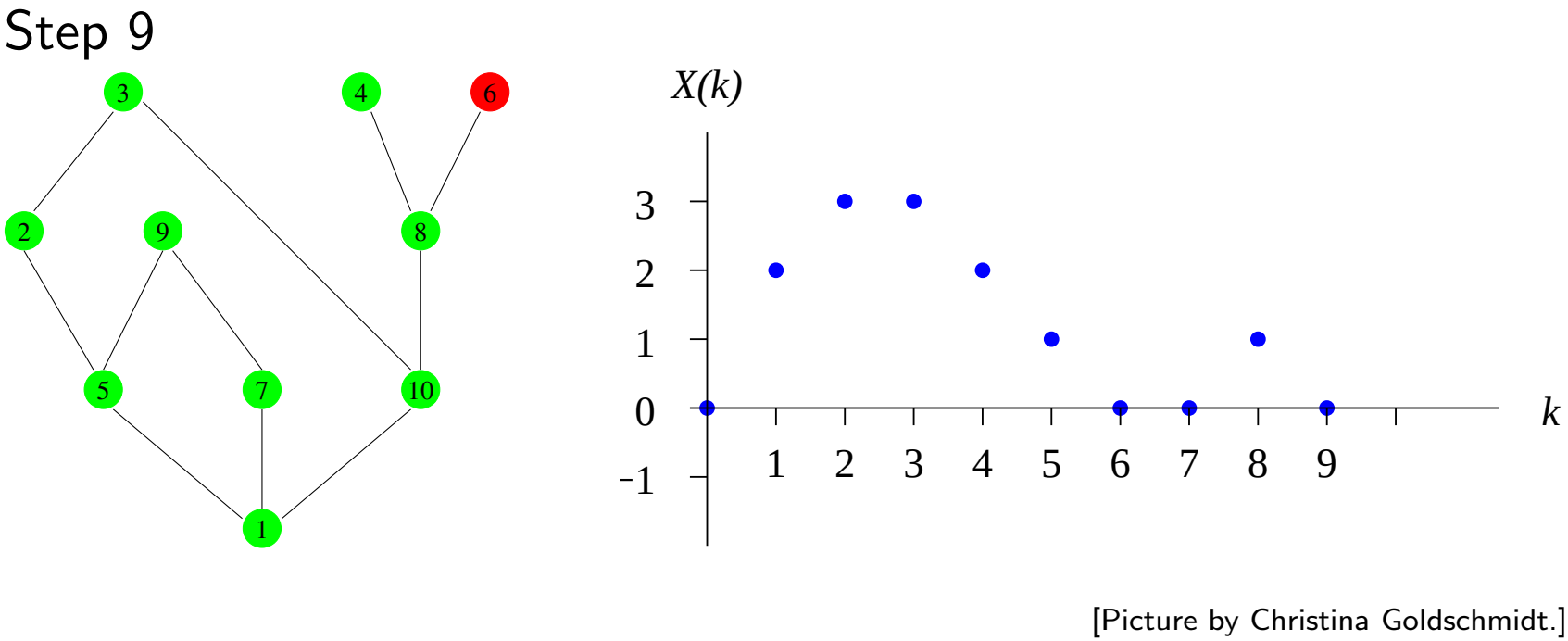
Depth-first walk



[Picture by Christina Goldschmidt.]

Current: 4 Alive: 6 Dead: 1,5,2,3,9,7,10,8

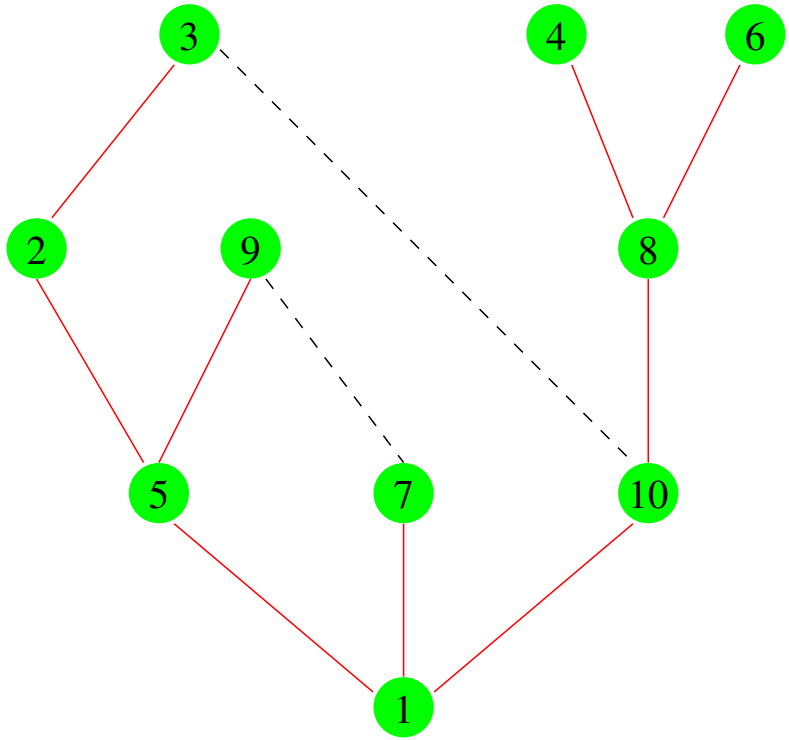
Depth-first walk



[Picture by Christina Goldschmidt.]

Current: 6 Alive: none Dead: 1,5,2,3,9,7,10,8,4

Depth-first tree



We essentially explored this underlying tree (the dashed edges made no difference to the depth-first walk). This tree is the depth-first tree associated with the graph.

Permitted edges

What are the set of graphs with a given depth-first tree/canonical discrete excursion?

Permitted edges

What are the set of graphs with a given depth-first tree/canonical discrete excursion?

In other words, where can we put surplus edges so that they don't change T?

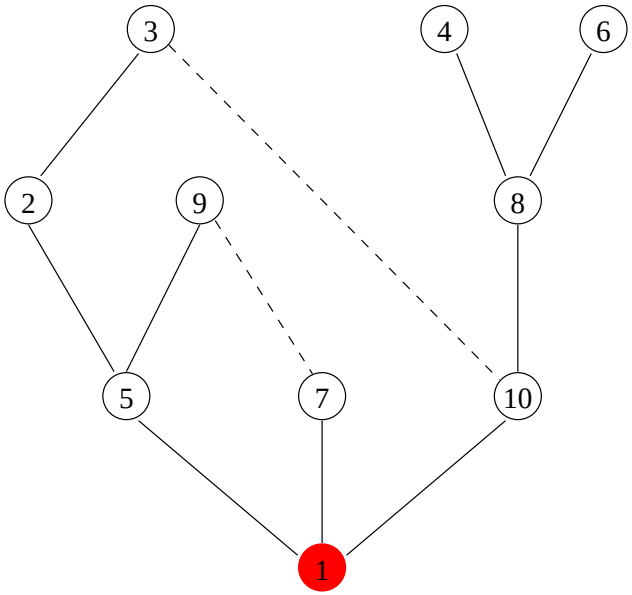
Permitted edges

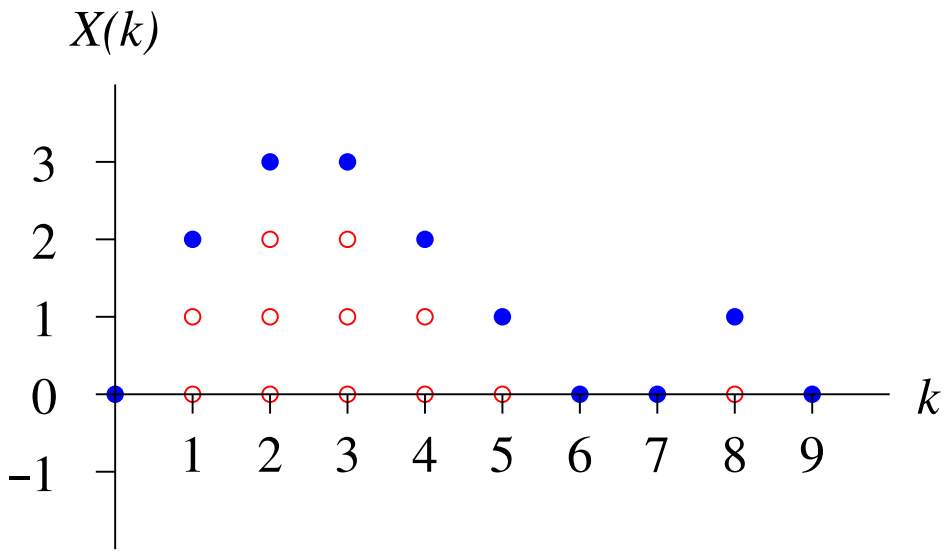
What are the set of graphs with a given depth-first tree/canonical discrete excursion?

In other words, where can we put surplus edges so that they don't change T?

Call such edges permitted.

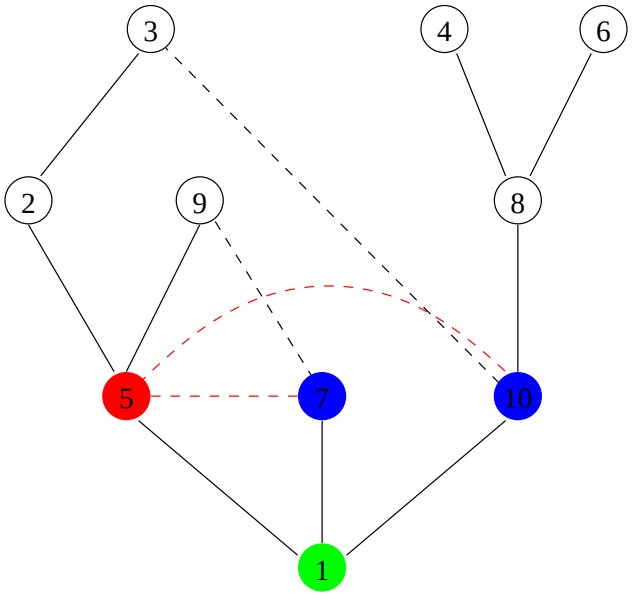
Depth-first walk and permitted edges

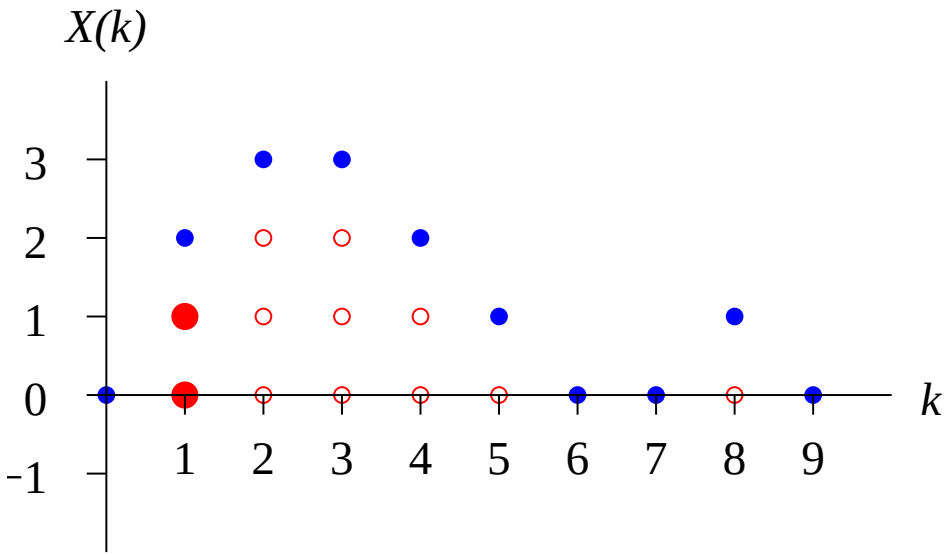




Step 0: $X(0) = 0$.

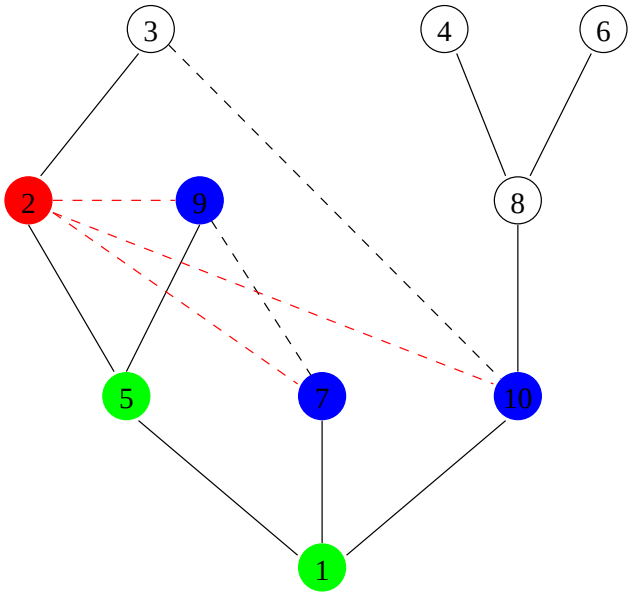
Depth-first walk and permitted edges

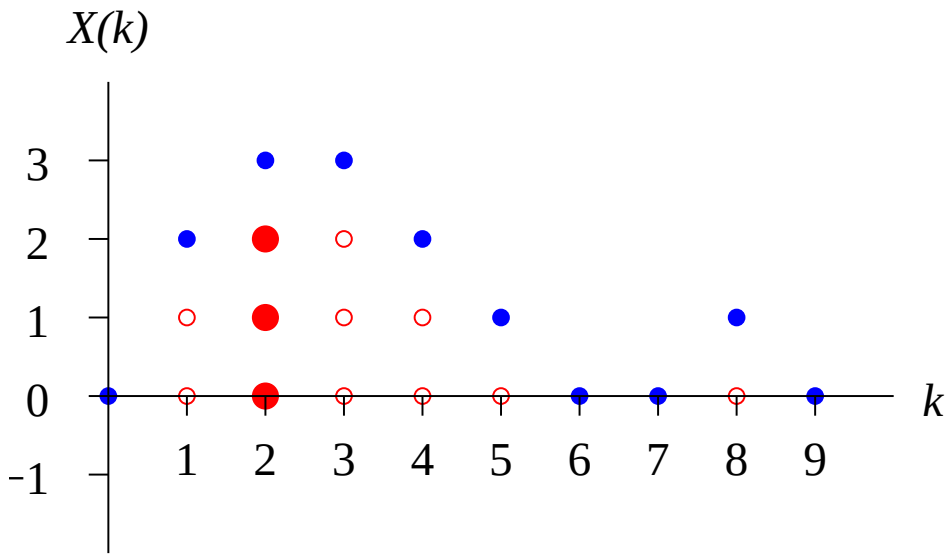




Step 1: $X(1) = 2$.

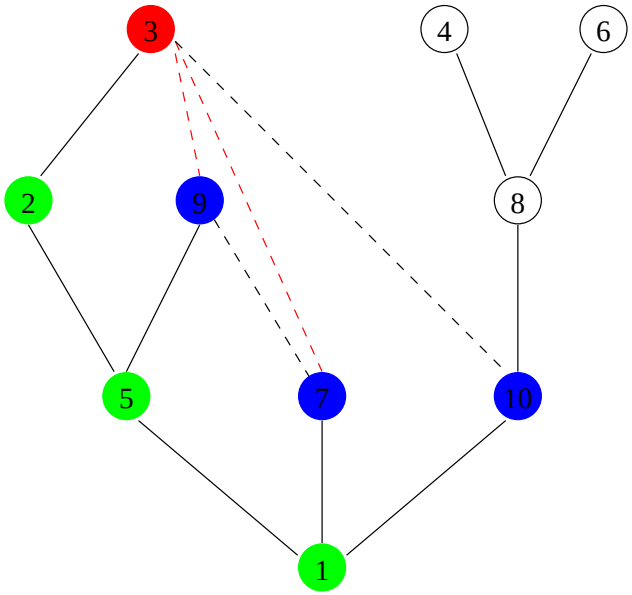
Depth-first walk and permitted edges

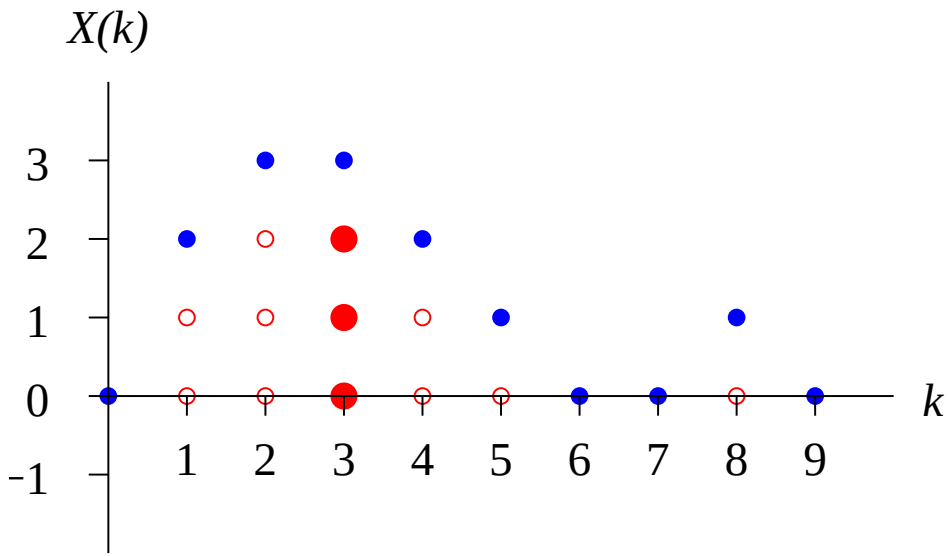




Step 2: $X(2) = 3$.

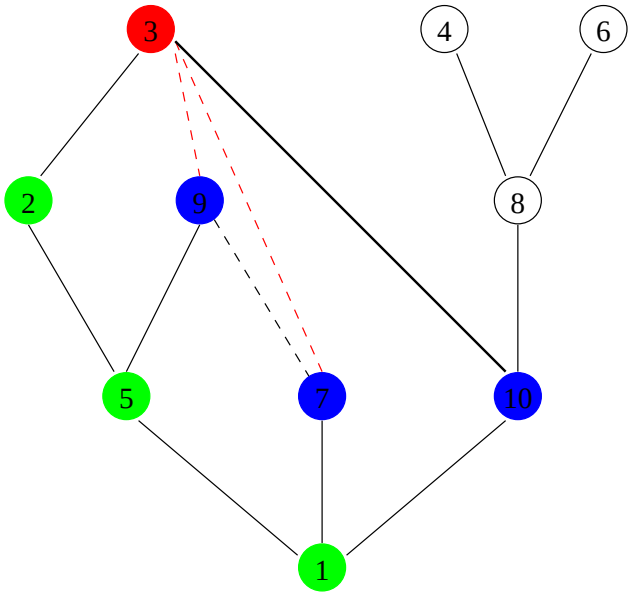
Depth-first walk and permitted edges

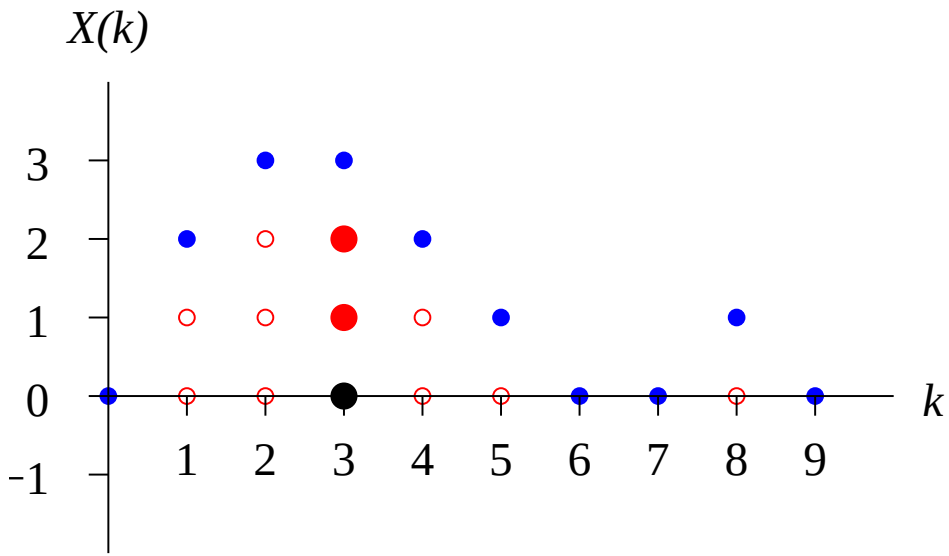




Step 3: $X(3) = 3$.

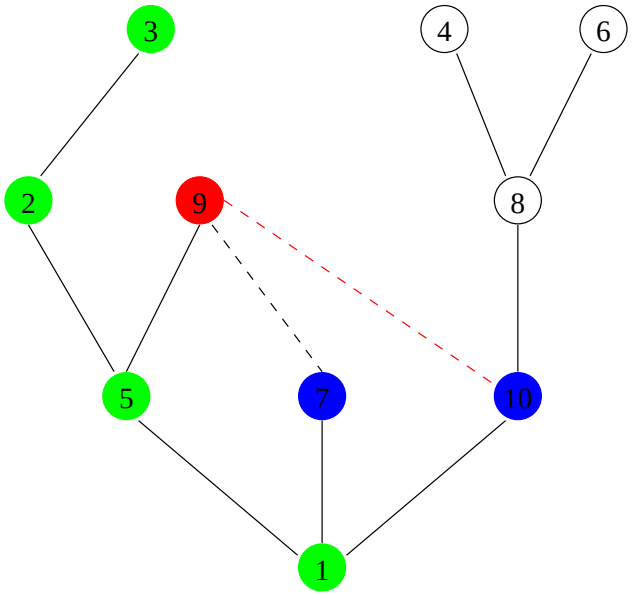
Depth-first walk and permitted edges

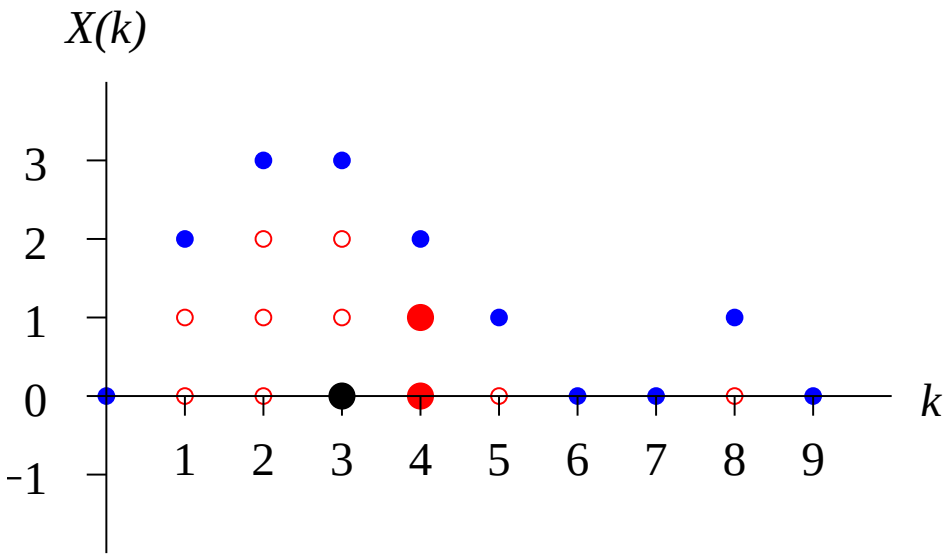




Step 3: $X(3) = 3$.

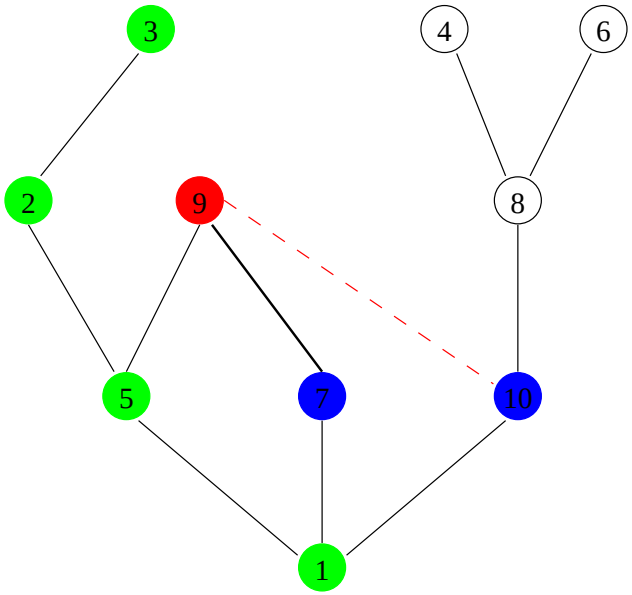
Depth-first walk and permitted edges

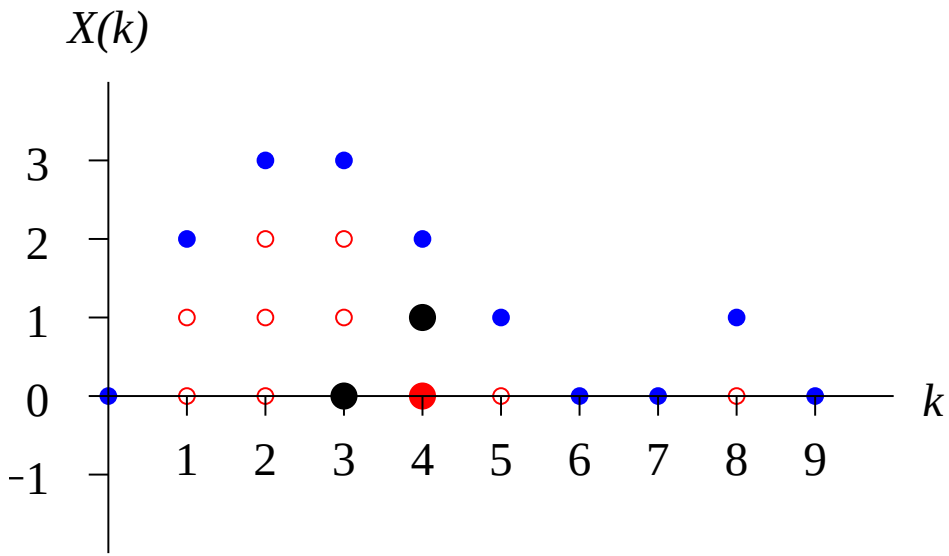




Step 4: $X(4) = 2$.

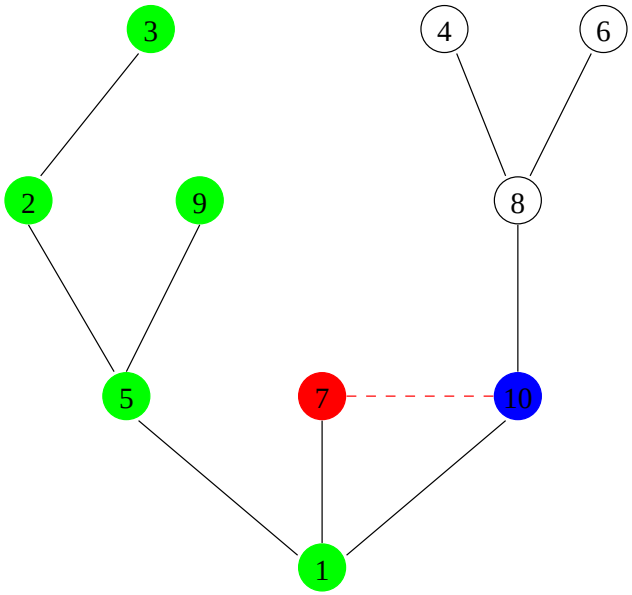
Depth-first walk and permitted edges

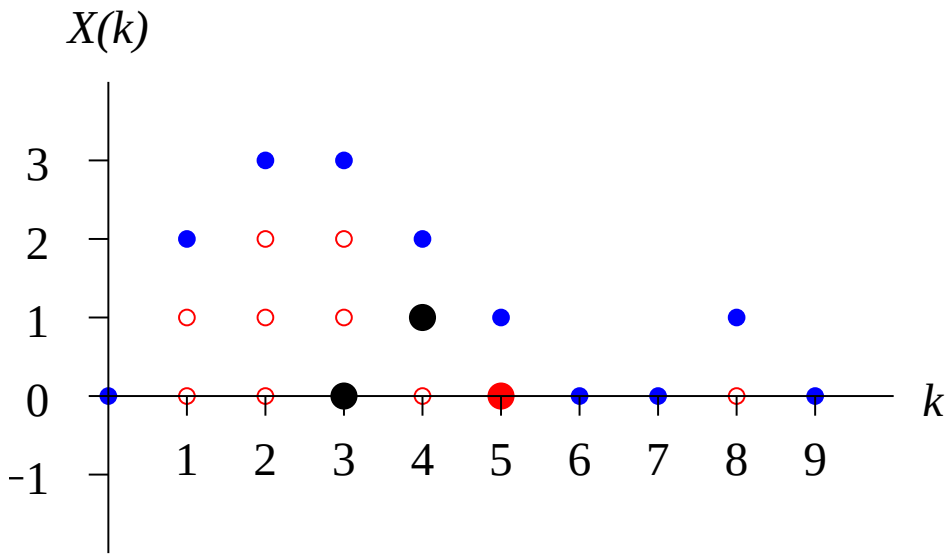




Step 4: $X(4) = 2$.

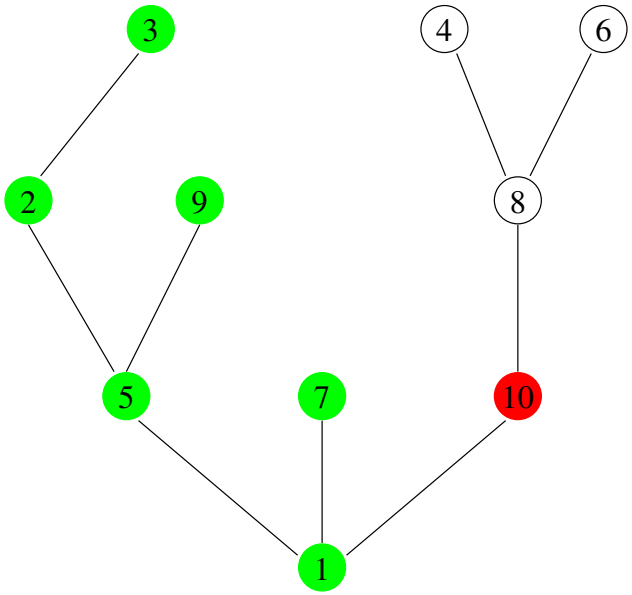
Depth-first walk and permitted edges

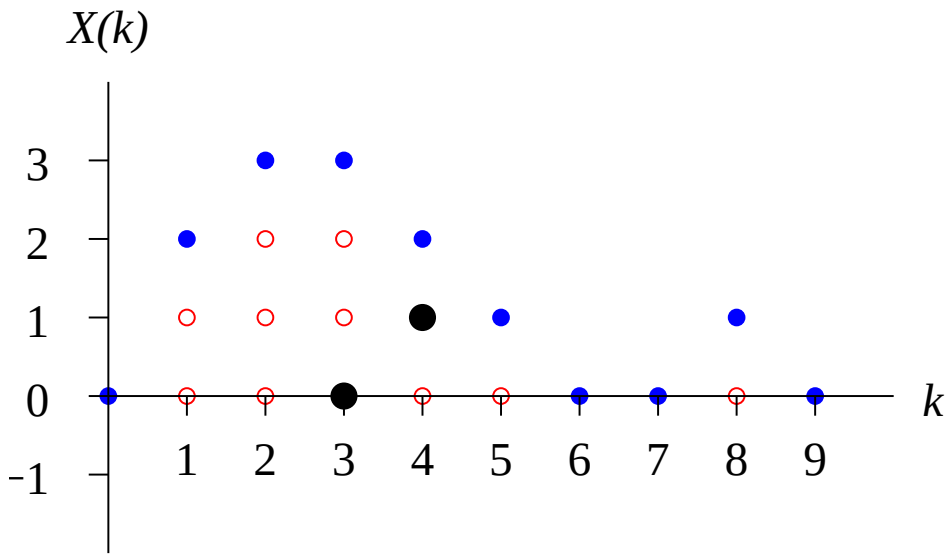




Step 5: $X(5) = 1$.

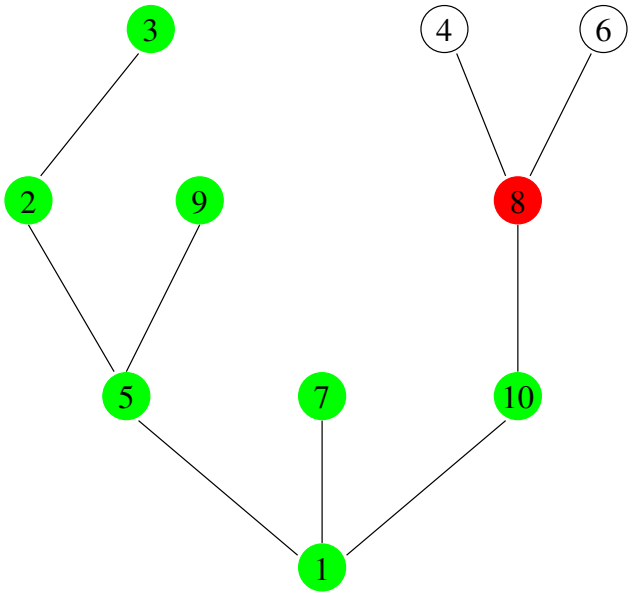
Depth-first walk and permitted edges

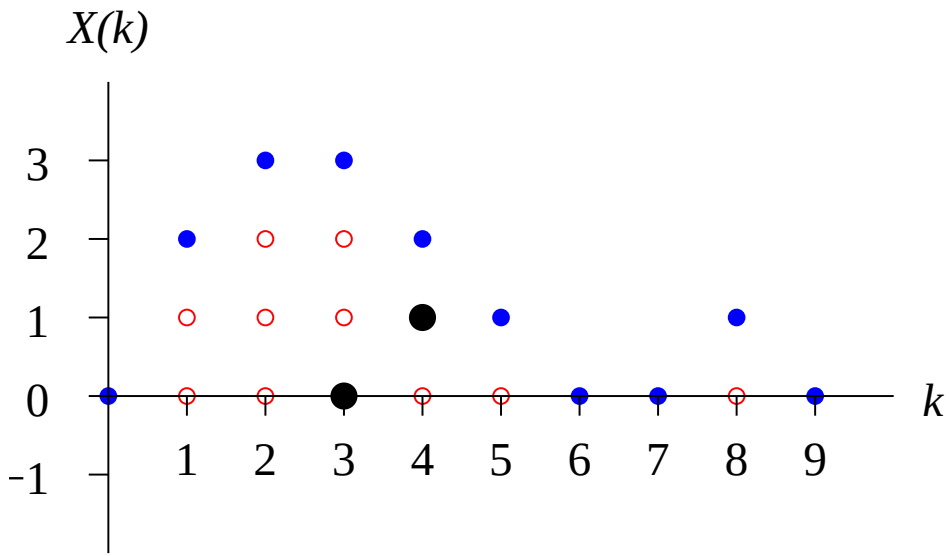




Step 6: $X(6) = 0$.

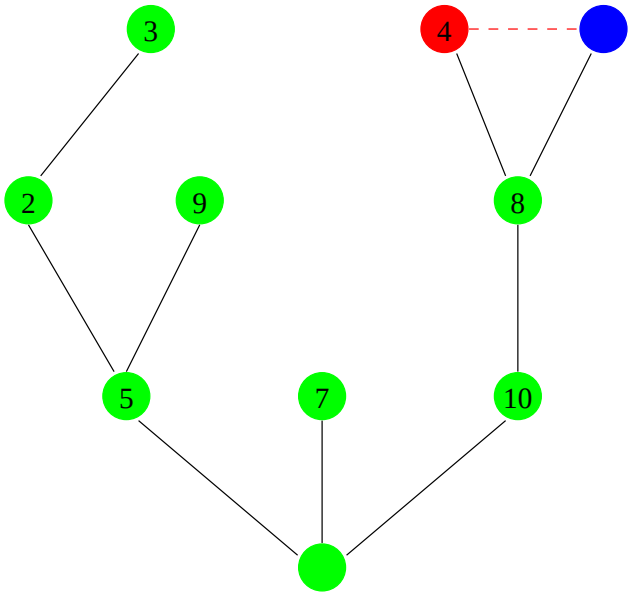
Depth-first walk and permitted edges

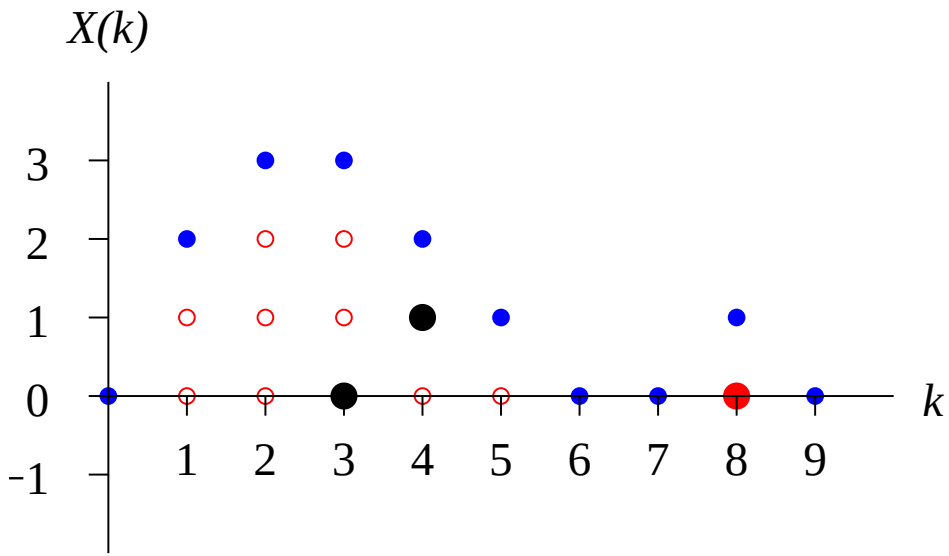




Step 7: $X(7) = 0$.

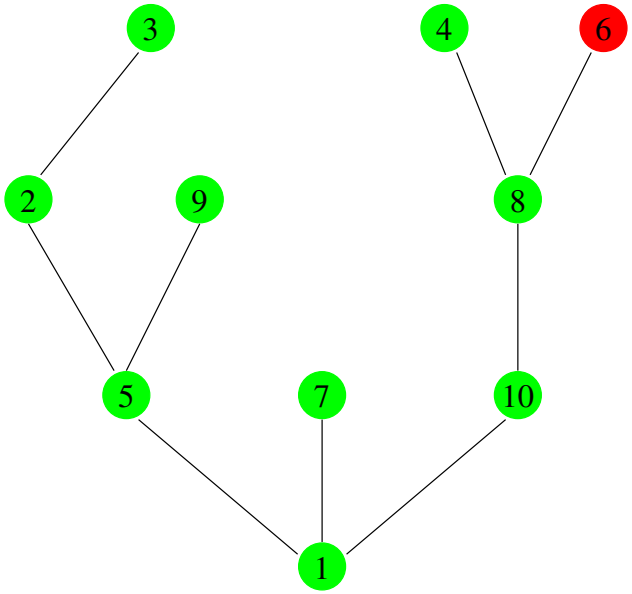
Depth-first walk and permitted edges

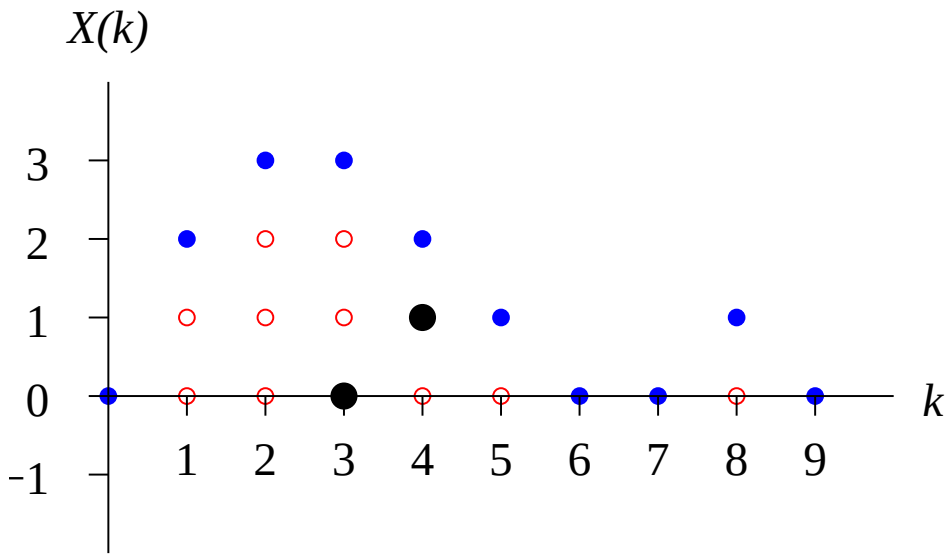




Step 8: $X(8) = 1$.

Depth-first walk and permitted edges





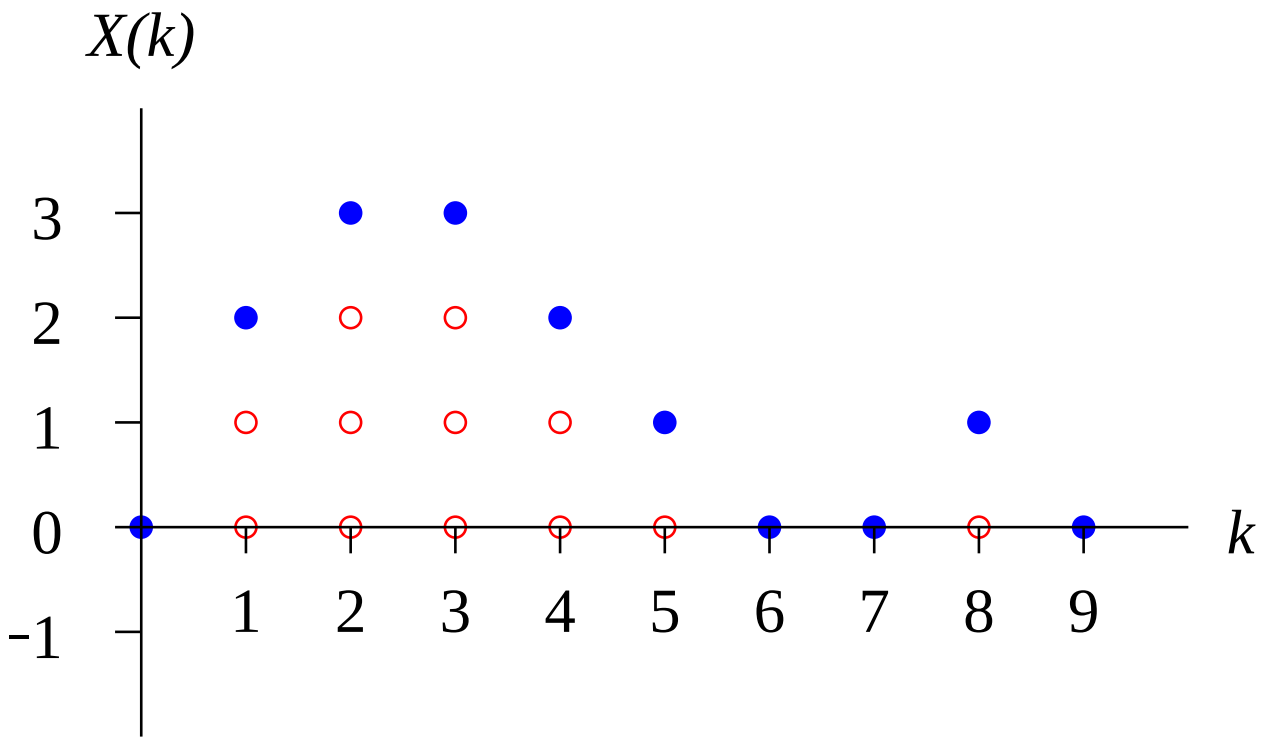
Step 10: $X(9) = 0$.

Area

At step $k \geq 0$ there are $X(k)$ permitted edges. So the total number is

$$a(T) = \sum_{k=0}^{m-1} X(k).$$

We call this the area of T.



Classifying graphs by depth-first tree

Let $\mathbb{G}_T$ be the set of graphs G such that $T(G) = T$. It follows that $|\mathbb{G}_T| = 2^{a(T)}$, since each permitted edge may either be included or not.

Classifying graphs by depth-first tree

Let $\mathbb{G}_T$ be the set of graphs G such that $T(G) = T$. It follows that $|\mathbb{G}_T| = 2^{a(T)}$, since each permitted edge may either be included or not.
Recall that $\mathbb{T}_{[m]}$ is the set of trees with label-set $[m] = \{1, 2, \ldots, m\}$. Then

$$\left\{\mathbb{G}_T : T \in \mathbb{T}_{[m]}\right\}$$

is a partition of the set of connected graphs on $[m]$.

Classifying graphs by depth-first tree

Let $\mathbb{G}_T$ be the set of graphs G such that $T(G) = T$. It follows that $|\mathbb{G}_T| = 2^{a(T)}$, since each permitted edge may either be included or not.
Recall that $\mathbb{T}_{[m]}$ is the set of trees with label-set $[m] = \{1, 2, \ldots, m\}$. Then

$$\left\{\mathbb{G}_T : T \in \mathbb{T}_{[m]}\right\}$$

is a partition of the set of connected graphs on $[m]$.
Furthermore, for given T, an element of $\mathbb{G}_T$ corresponds to a set of marks (with integer coordinates) under the discrete excursion for T.

Generating a random connected graph

We wanted a way of generating a uniform random connected graph with s surplus edges, a discrete analog of the limiting "excursion plus marks" picture.

Generating a random connected graph

We wanted a way of generating a uniform random connected graph with s surplus edges, a discrete analog of the limiting "excursion plus marks" picture.

- First select a tree randomly by picking each of the labelled trees on $1, 2, \ldots, m$ with a probability weight proportional to $2^{a(T)}$.

Generating a random connected graph

We wanted a way of generating a uniform random connected graph with s surplus edges, a discrete analog of the limiting "excursion plus marks" picture.

- First select a tree randomly by picking each of the labelled trees on $1, 2, \ldots, m$ with a probability weight proportional to $2^{a(T)}$.
- Then add each of the $a(T)$ possible (allowed) surplus edges independently with probability $1/2$.

Generating a random connected graph

We wanted a way of generating a uniform random connected graph with s surplus edges, a discrete analog of the limiting "excursion plus marks" picture.

- First select a tree randomly by picking each of the labelled trees on $1, 2, \ldots, m$ with a probability weight proportional to $2^{a(T)}$.
- Then add each of the $a(T)$ possible (allowed) surplus edges independently with probability $1/2$. In other words, pick a random set of marks under the depth-first walk of T, each mark independently present with probability $1/2$.

Generating a random connected graph

We wanted a way of generating a uniform random connected graph with s surplus edges, a discrete analog of the limiting "excursion plus marks" picture.

- First select a tree randomly by picking each of the labelled trees on $1, 2, \ldots, m$ with a probability weight proportional to $2^{a(T)}$.
- Then add each of the $a(T)$ possible (allowed) surplus edges independently with probability $1/2$. In other words, pick a random set of marks under the depth-first walk of T, each mark independently present with probability $1/2$.

Conditional on obtaining s surplus edges, the tree thus generated is a uniform random connected graph on m vertices with s surplus edges.

Tilting

Unconditionally, the above method generates a uniformly random connected graph on $1, \ldots, m$. When the desired surplus s is far from $m^2/2$, the conditioning is extremely unlikely. We can modify the method to make it more likely we get the desired surplus.

Tilting

Unconditionally, the above method generates a uniformly random connected graph on $1, \ldots, m$. When the desired surplus s is far from $m^2/2$, the conditioning is extremely unlikely. We can modify the method to make it more likely we get the desired surplus.

- Fix any $p \in [0, 1)$.

Tilting

Unconditionally, the above method generates a uniformly random connected graph on $1, \ldots, m$. When the desired surplus s is far from $m^2/2$, the conditioning is extremely unlikely. We can modify the method to make it more likely we get the desired surplus.

- Fix any $p \in [0, 1)$.
- First select a tree randomly by picking each of the labelled trees on $1, 2, \ldots, m$ with a probability weight proportional to $(1-p)^{-a(T)}$.

Tilting

Unconditionally, the above method generates a uniformly random connected graph on $1, \ldots, m$. When the desired surplus s is far from $m^2/2$, the conditioning is extremely unlikely. We can modify the method to make it more likely we get the desired surplus.

- Fix any $p \in [0, 1)$.
- First select a tree randomly by picking each of the labelled trees on $1, 2, \ldots, m$ with a probability weight proportional to $(1-p)^{-a(T)}$.
- Then add each of the $a(T)$ possible (allowed) surplus edges independently with probability p.

Tilting

Unconditionally, the above method generates a uniformly random connected graph on $1, \ldots, m$. When the desired surplus s is far from $m^2/2$, the conditioning is extremely unlikely. We can modify the method to make it more likely we get the desired surplus.

- Fix any $p \in [0, 1)$.
- First select a tree randomly by picking each of the labelled trees on $1, 2, \ldots, m$ with a probability weight proportional to $(1-p)^{-a(T)}$.
- Then add each of the $a(T)$ possible (allowed) surplus edges independently with probability p.

For $p > 0$ still have: conditional on obtaining s surplus edges, the tree thus generated is a uniform random connected graph on m vertices with s surplus edges.

Tilting

Unconditionally, the above method generates a uniformly random connected graph on $1, \ldots, m$. When the desired surplus s is far from $m^2/2$, the conditioning is extremely unlikely. We can modify the method to make it more likely we get the desired surplus.

- Fix any $p \in [0, 1)$.
- First select a tree randomly by picking each of the labelled trees on $1, 2, \ldots, m$ with a probability weight proportional to $(1-p)^{-a(T)}$.
- Then add each of the $a(T)$ possible (allowed) surplus edges independently with probability p.

For $p > 0$ still have: conditional on obtaining s surplus edges, the tree thus generated is a uniform random connected graph on m vertices with s surplus edges.
For $p = 0$ we are just choosing a uniformly random tree.

Tilting.

We wanted a way of generating a random component of $G_{n,1/n}$ with size m. We are now in a position to do this.

Tilting.

We wanted a way of generating a random component of $G_{n,1/n}$ with size m. We are now in a position to do this.

- First select a tree randomly by picking each of the labelled trees T on $1, 2, \ldots, m$ with a probability weight proportional to $(1 - 1/n)^{-a(T)}$.

Tilting.

We wanted a way of generating a random component of $G_{n,1/n}$ with size m. We are now in a position to do this.

- First select a tree randomly by picking each of the labelled trees T on $1, 2, \ldots, m$ with a probability weight proportional to $(1 - 1/n)^{-a(T)}$.
- Having chosen T, then add each of the $a(T)$ allowed surplus edges (marks) independently with probability $1/n$.

Tilting.

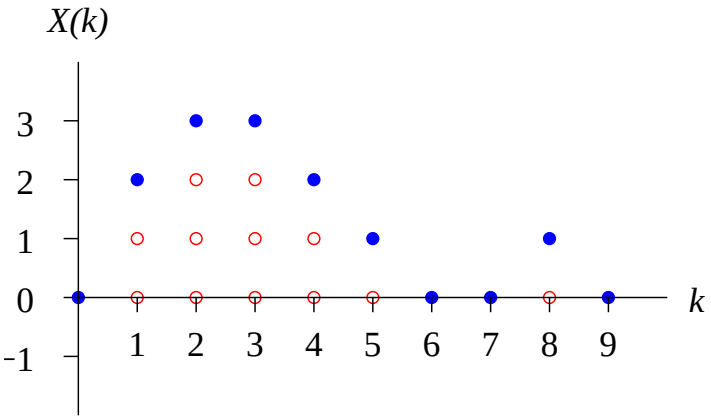
We wanted a way of generating a random component of $G_{n,1/n}$ with size m. We are now in a position to do this.

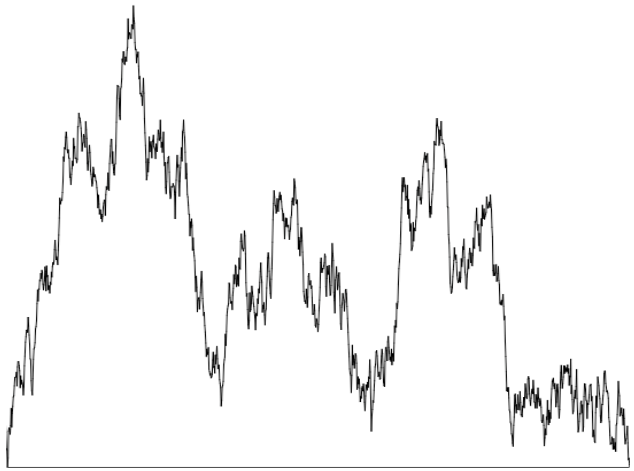
- First select a tree randomly by picking each of the labelled trees T on $1, 2, \ldots, m$ with a probability weight proportional to $(1 - 1/n)^{-a(T)}$.
- Having chosen T, then add each of the $a(T)$ allowed surplus edges (marks) independently with probability $1/n$.

The result: a uniformly random connected component of $G_{n,1/n}$, conditioned upon having size m.

The scaling limit of depth-first walks.

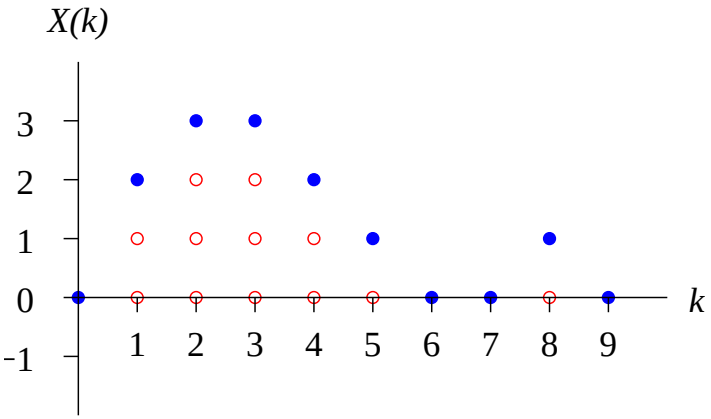
When time is scaled by $1/m$, space is scaled by $1/\sqrt{m}$, the limit of depth-first walk of a *uniform* tree is *Brownian excursion*.

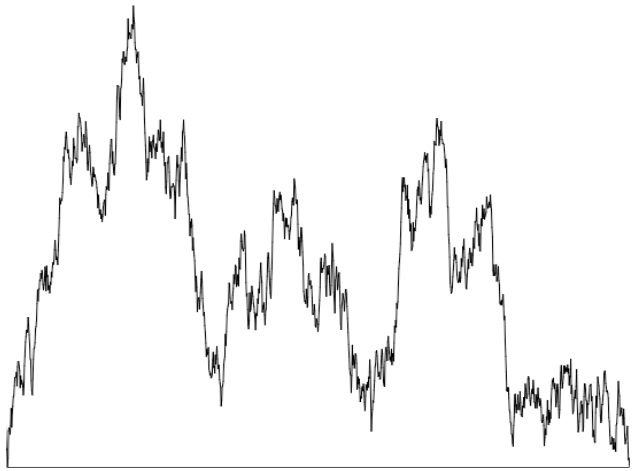




The scaling limit of depth-first walks.

When time is scaled by $1/m$, space is scaled by $1/\sqrt{m}$, the limit of depth-first walk of a *uniform* tree is *Brownian excursion*.

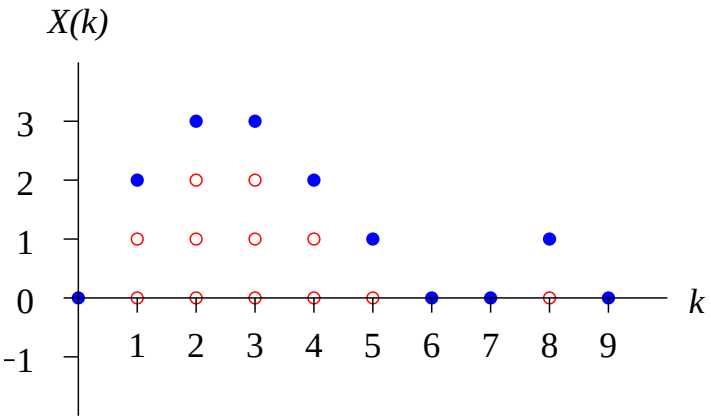


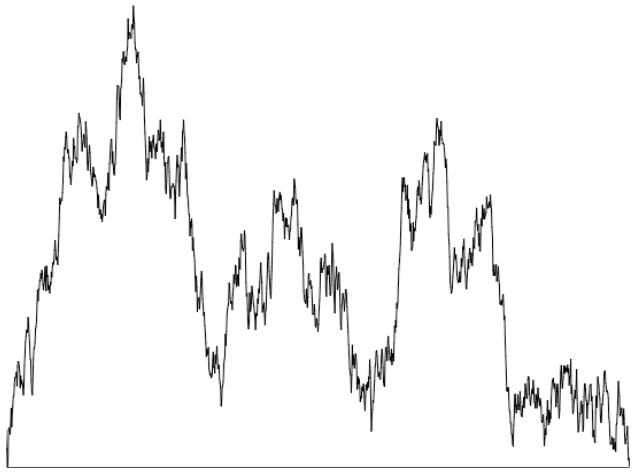


The *area* $a(T)$ is then $\Theta(m^{3/2})$, so if $m = n^{2/3}$ then $a(T) = \Theta(n)$ and

$$\left(1 - \frac{1}{n}\right)^{-a(T)} \sim e^{-a(T)/n}.$$

The scaling limit of depth-first walks.

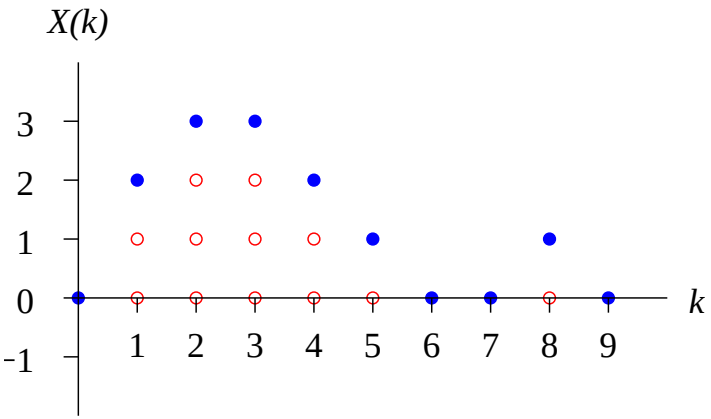


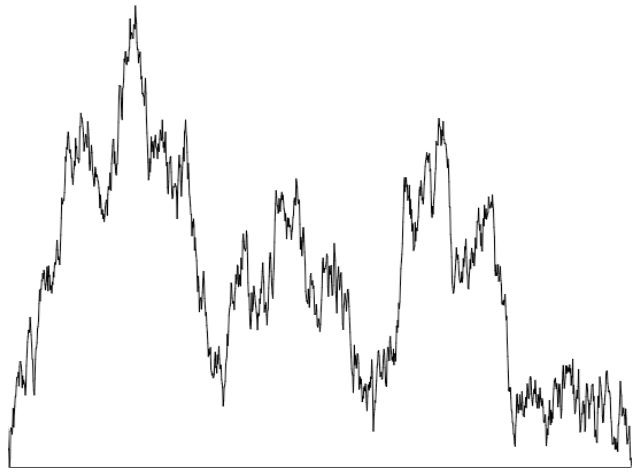


$$\left(1-\frac{1}{n}\right)^{-a(T)} \sim e^{-a(T)/n}.$$

We choose a tree with probability prop. to $(1-1/n)^{-a(T)}$.

The scaling limit of depth-first walks.



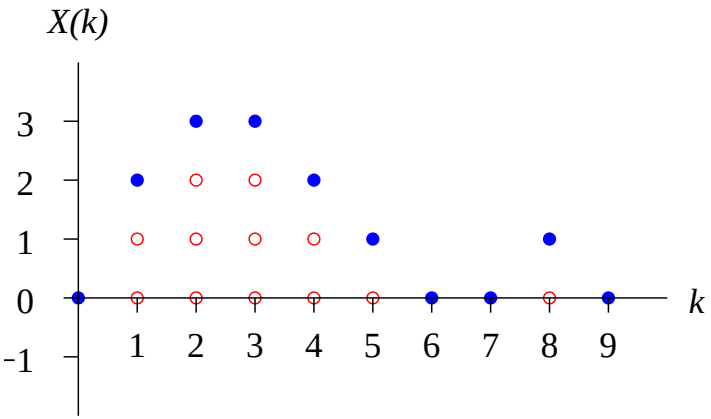


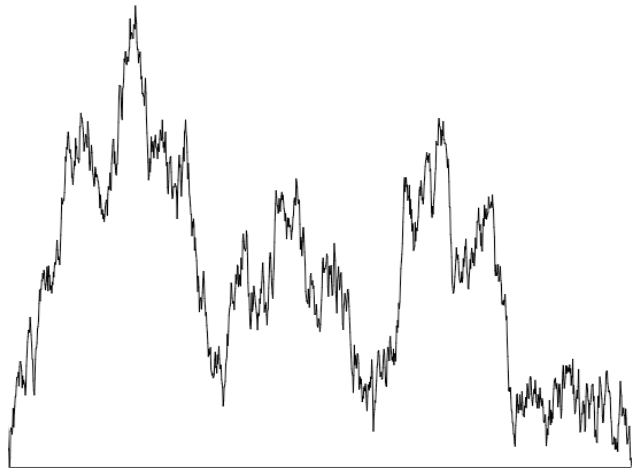
$$\left(1-\frac{1}{n}\right)^{-a(T)} \sim e^{-a(T)/n}.$$

We choose a tree with probability prop. to $(1-1/n)^{-a(T)}$.

This corresponds to an *exponential* "tilt" of the distribution of the limiting Brownian excursion.

The scaling limit of depth-first walks.





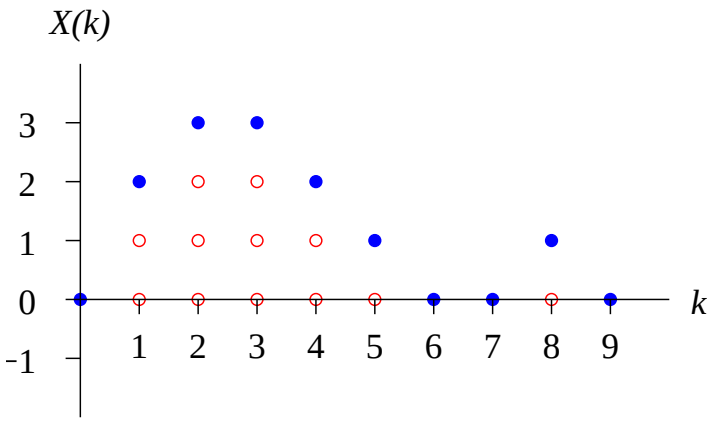
$$\left(1-\frac{1}{n}\right)^{-a(T)} \sim e^{-a(T)/n}.$$

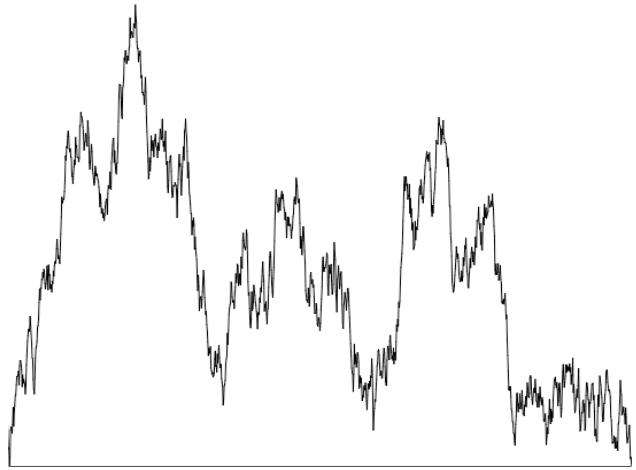
We choose a tree with probability prop. to $(1-1/n)^{-a(T)}$.

This corresponds to an *exponential* "tilt" of the distribution of the limiting Brownian excursion.

This tilt can be shown to be exactly the effect of the quadratic drift. (Using Girsanov's theorem.)

The scaling limit of depth-first walks.





$$\left(1-\frac{1}{n}\right)^{-a(T)} \sim e^{-a(T)/n}.$$

In the limit, the binomial point process of marks under the depth-first walk becomes a Poisson process of marks under the excursion.

Limit for a component of $G_{n,1/n}$

A component of size about $cn^{2/3}$ has distributional limit:

Limit for a component of $G_{n,1/n}$

A component of size about $cn^{2/3}$ has distributional limit:

- Tree coded by a Brownian excursion of length c, with exponential tilt; plus

Limit for a component of $G_{n,1/n}$

A component of size about $cn^{2/3}$ has distributional limit:

- Tree coded by a Brownian excursion of length c, with exponential tilt; plus
- identifications given by Poisson random points under the excursion.

Limit for a component of $G_{n,1/n}$

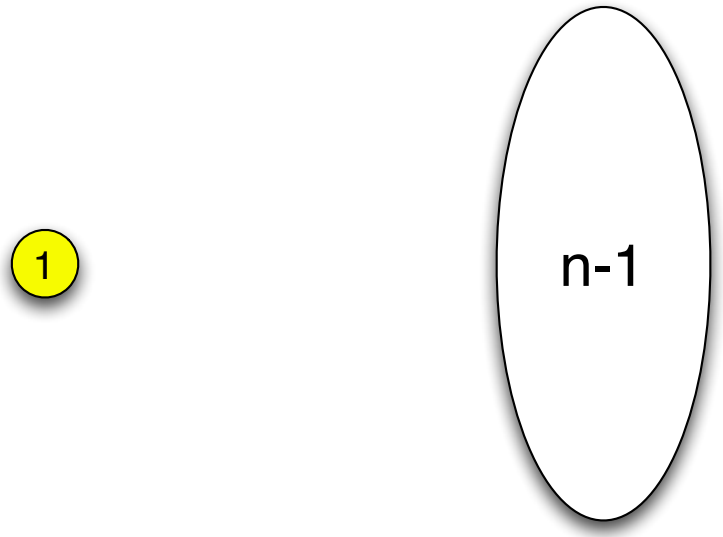
A component of size about $cn^{2/3}$ has distributional limit:

- Tree coded by a Brownian excursion of length c, with exponential tilt; plus
- identifications given by Poisson random points under the excursion.

Now have an asymptotic description of the structure of each "macroscopic" component of the critical random graph.

Brownian motion with drift?

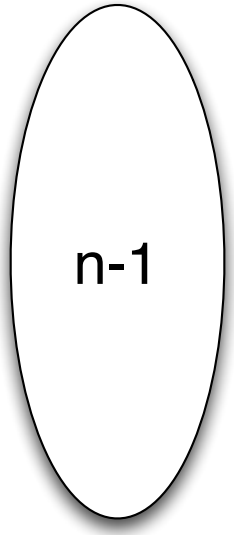
Mean number of neighbours:



Brownian motion with drift?

Mean number of neighbours:

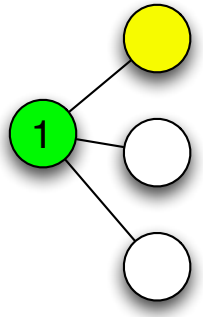
- $\mathrm{Bin}(n-1, 1/n) = 1 - 1/n$ at step 1.

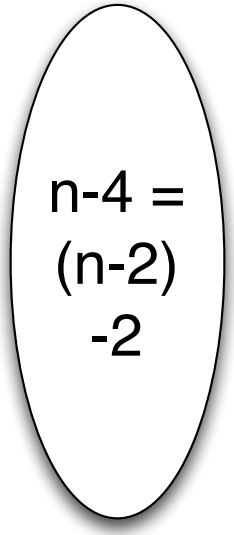


Brownian motion with drift?

Mean number of neighbours:

- $\mathrm{Bin}(n-1, 1/n) = 1 - 1/n$ at step 1.
- $1 - 4/n$ at step 2.

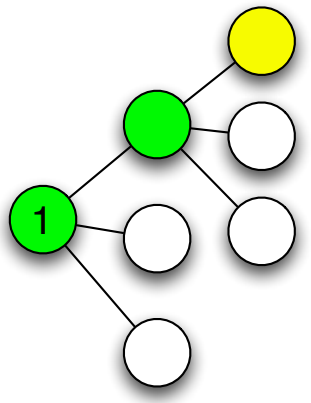


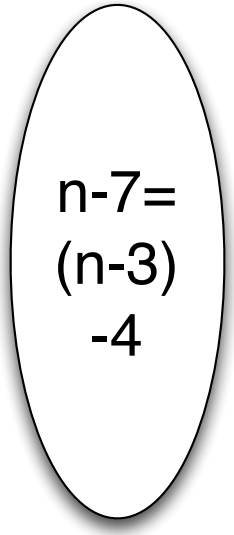


Brownian motion with drift?

Mean number of neighbours:

- $\mathrm{Bin}(n-1, 1/n) = 1 - 1/n$ at step 1.
- $1 - 4/n$ at step 2.
- $1 - 7/n$ at step 3.

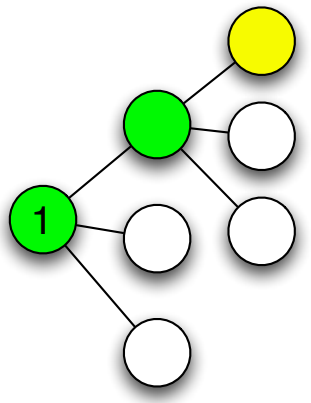


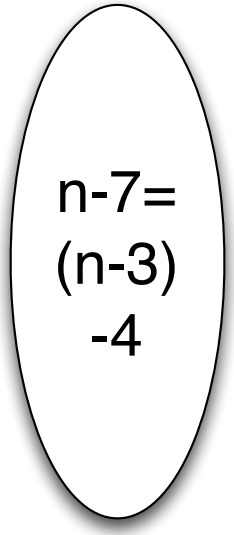


Brownian motion with drift?

Mean number of neighbours:

- $\mathrm{Bin}(n-1, 1/n) = 1 - 1/n$ at step 1.
- $1 - 4/n$ at step 2.
- $1 - 7/n$ at step 3.



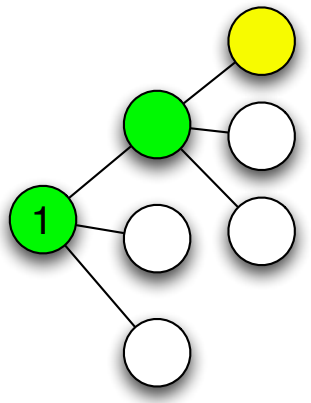


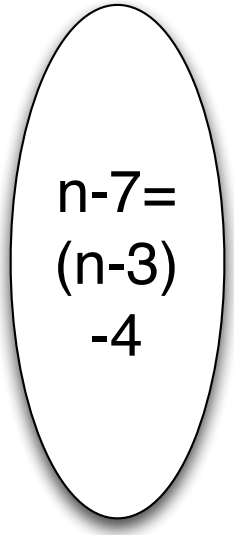
- At step i: infinitesimal drift
  $-i/n - (\#\text{ of active vertices})/n = -(i + O(\sqrt{i}))/n$.

Brownian motion with drift?

Mean number of neighbours:

- $\mathrm{Bin}(n-1, 1/n) = 1 - 1/n$ at step 1.
- $1 - 4/n$ at step 2.
- $1 - 7/n$ at step 3.



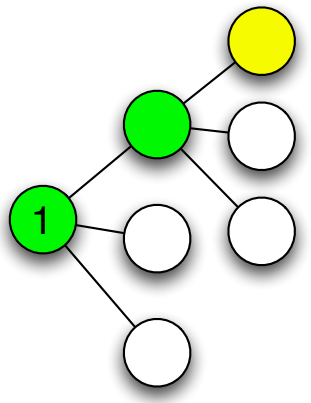


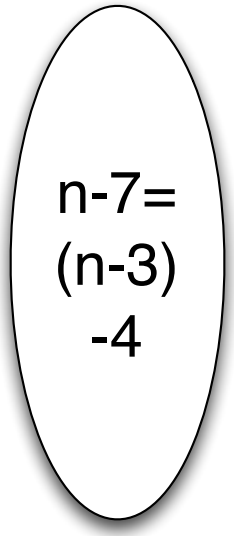
- At step i: infinitesimal drift $-i/n - (\#\text{ of active vertices})/n = -(i + O(\sqrt{i}))/n$.
- Accumulated drift about $-i^2/2n$, or $-n^{1/3}$ at time $n^{2/3}$.

Brownian motion with drift?

Mean number of neighbours:

- $\mathrm{Bin}(n-1, 1/n) = 1 - 1/n$ at step 1.
- $1 - 4/n$ at step 2.
- $1 - 7/n$ at step 3.





- At step i: infinitesimal drift $-i/n - (\#\text{ of active vertices})/n = -(i + O(\sqrt{i}))/n$.
- Accumulated drift about $-i^2/2n$, or $-n^{1/3}$ at time $n^{2/3}$.
- The first component is explored when discrete walk hits -1; start again from 0 and explore another component! This is why reflection at running infimum.

Need to know

- What the MSTs of these components look like.

Need to know

- What the MSTs of these components look like.
- How these MSTs hook together.

How the MSTs hook together

List the components of $G_{n,1/n}$ as $C_1, C_2, \ldots$.

How the MSTs hook together

List the components of $G_{n,1/n}$ as $C_1, C_2, \ldots$.
Recall that in terms of their vertex sets, these are also the components of $T_{n,1/n}$.

How the MSTs hook together

List the components of $G_{n,1/n}$ as $C_1, C_2, \ldots$.
Recall that in terms of their vertex sets, these are also the components of $T_{n,1/n}$.

- The next edge that joints two components will join a given C_i and C_j with probability proportional to $|C_i| \cdot |C_j|$.

How the MSTs hook together

List the components of $G_{n,1/n}$ as $C_1, C_2, \ldots$.
Recall that in terms of their vertex sets, these are also the components of $T_{n,1/n}$.

- The next edge that joints two components will join a given C_i and C_j with probability proportional to $|C_i| \cdot |C_j|$.
- In the limit, the merging of components is governed by the multiplicative coalescent. (Aldous 1997.)

How the MSTs hook together

List the components of $G_{n,1/n}$ as $C_1, C_2, \ldots$.
Recall that in terms of their vertex sets, these are also the components of $T_{n,1/n}$.

- The next edge that joints two components will join a given C_i and C_j with probability proportional to $|C_i| \cdot |C_j|$.
- In the limit, the merging of components is governed by the multiplicative coalescent. (Aldous 1997.)
- Aside: it is a fair bit of work to define the multiplicative coalescent dynamics directly on spaces of sequences of measured metric spaces, and show that the resulting dynamics have nice closure and continuity properties.

How the MSTs hook together

List the components of $G_{n,1/n}$ as $C_1, C_2, \ldots$.
Recall that in terms of their vertex sets, these are also the components of $T_{n,1/n}$.

- The next edge that joints two components will join a given C_i and C_j with probability proportional to $|C_i| \cdot |C_j|$.
- In the limit, the merging of components is governed by the multiplicative coalescent. (Aldous 1997.)
- Aside: it is a fair bit of work (ongoing) to define the multiplicative coalescent dynamics directly on spaces of sequences of measured metric spaces, and show that the resulting dynamics have nice closure and continuity properties.

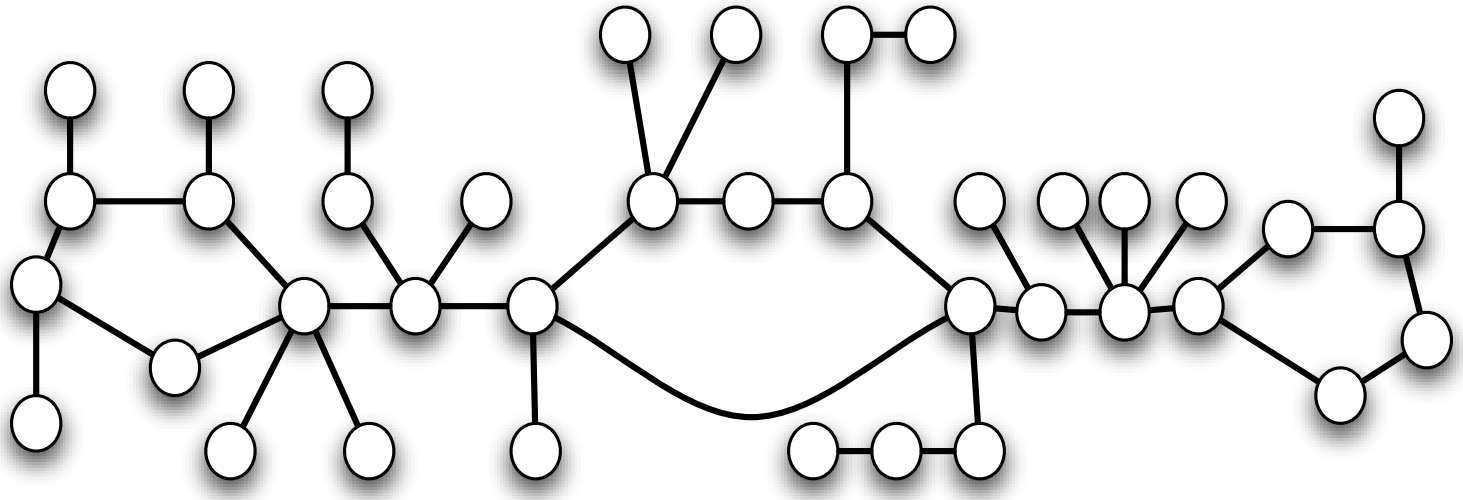
What the MSTs look like

- We will use the cycle breaking algorithm to figure this out.

What the MSTs look like

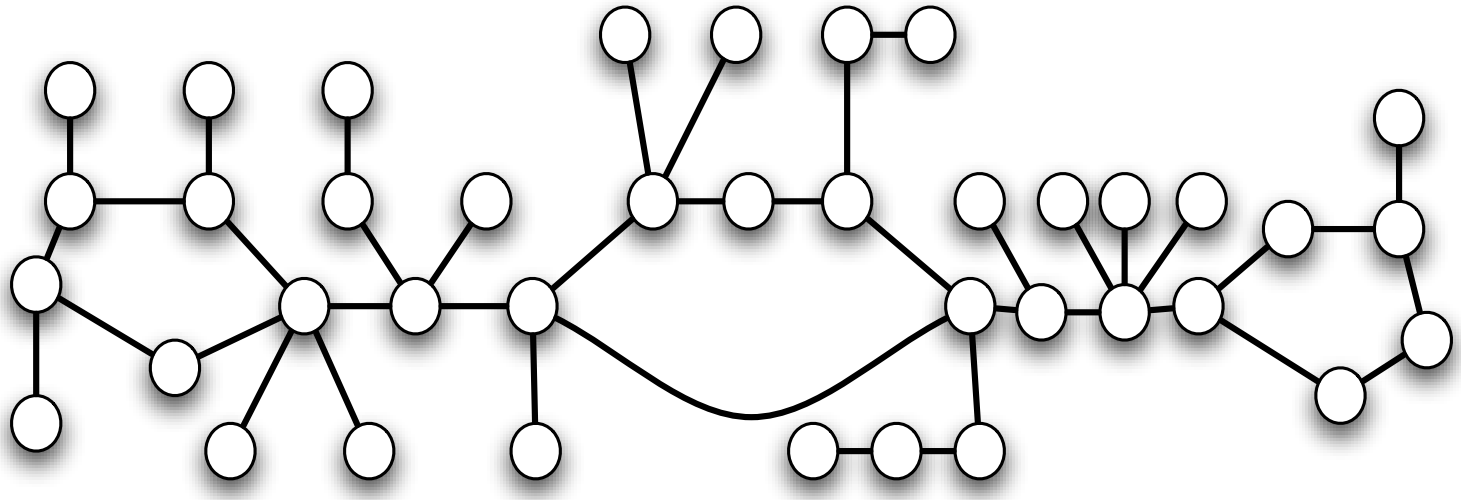
- We will use the cycle breaking algorithm to figure this out.
- First need to explain another description of a component of $G_{n,1/n}$, so we can better see the geometry (cycle structure).

The cycle structure of a connected graph.



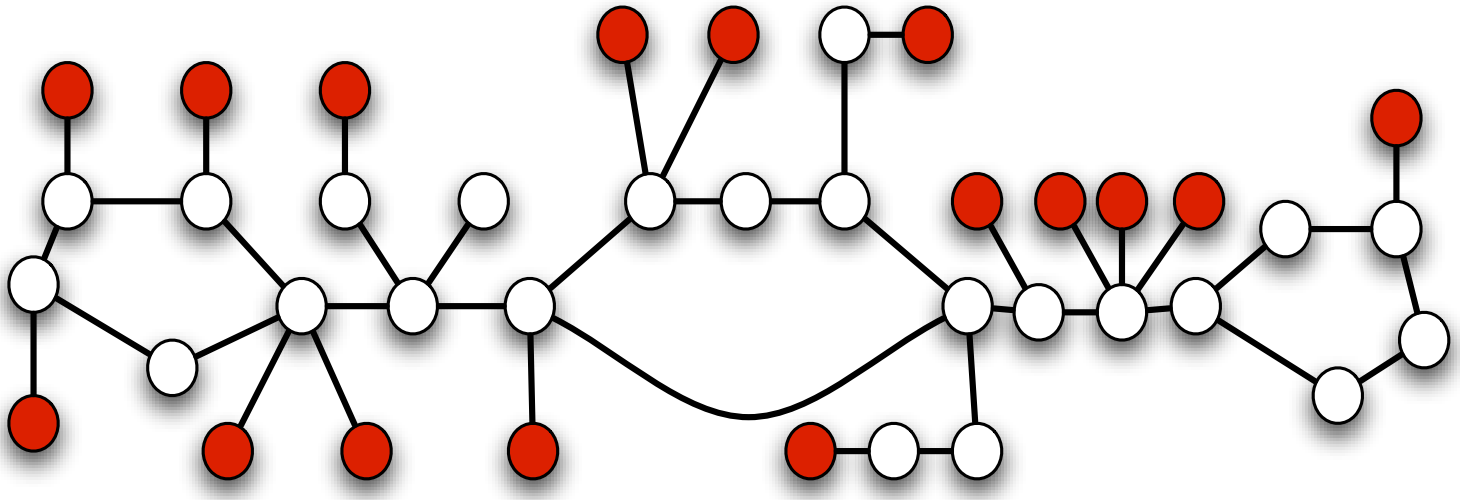
- Vertex labels suppressed so you can see what's going on.

The cycle structure of a connected graph.



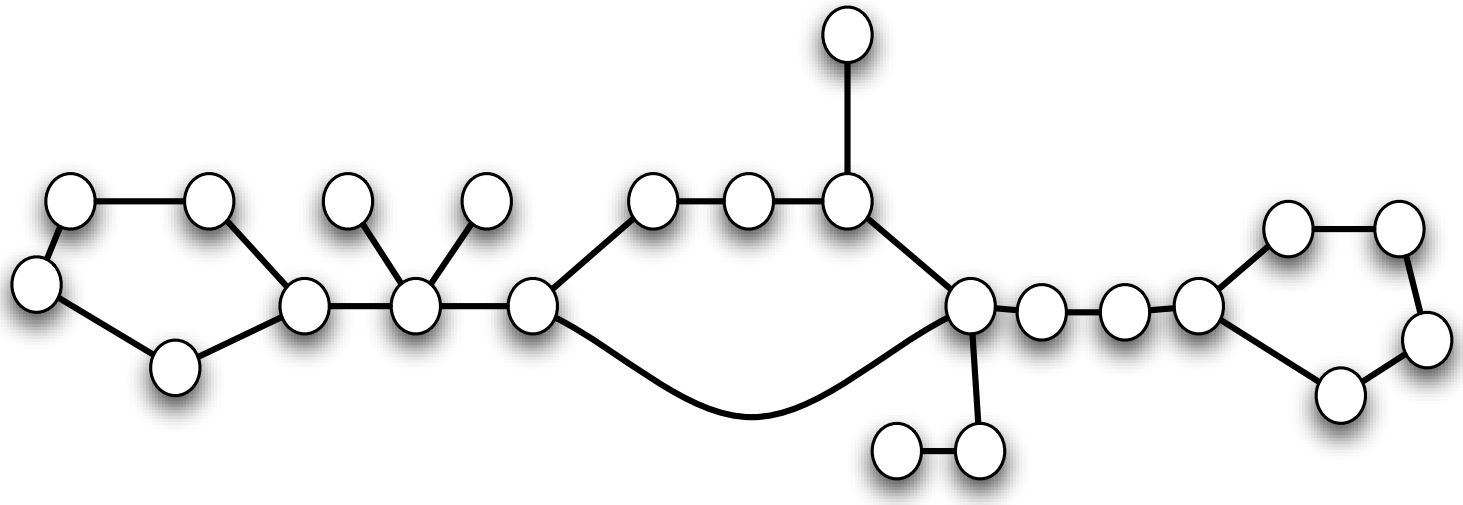
- Vertex labels suppressed so you can see what's going on.
- Leaves (nodes of degree 1) aren't in cycles, so don't affect the cycle structure – remove them!

The cycle structure of a connected graph.



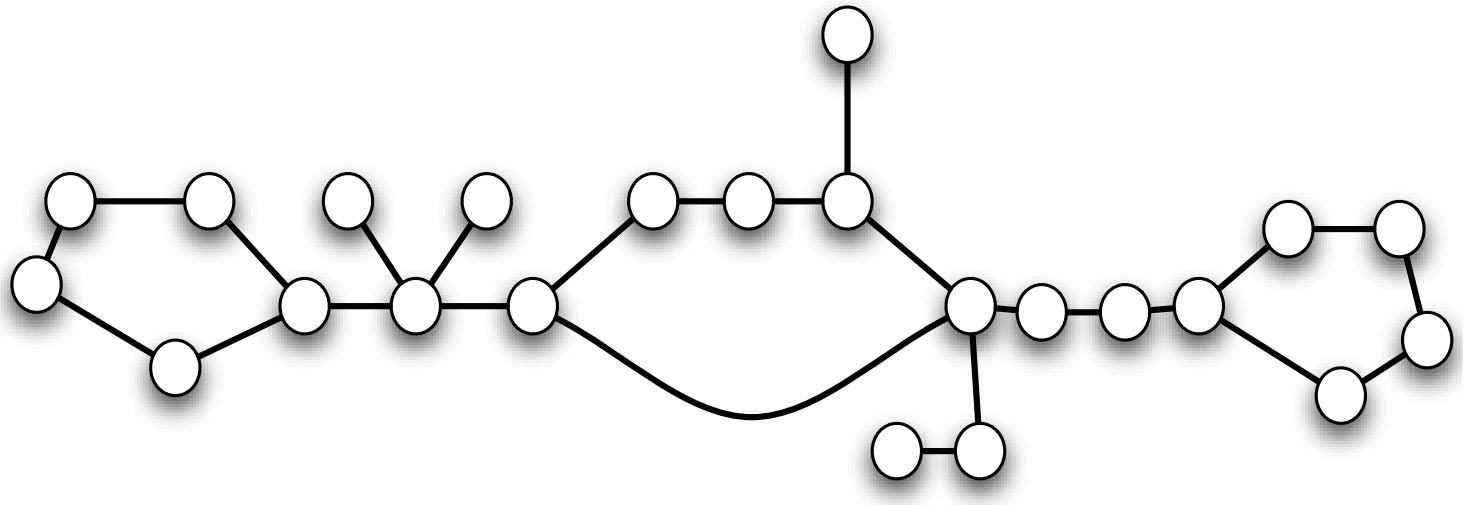
- Vertex labels suppressed so you can see what's going on.
- Leaves (nodes of degree 1) aren't in cycles, so don't affect the cycle structure – remove them!

The cycle structure of a connected graph.



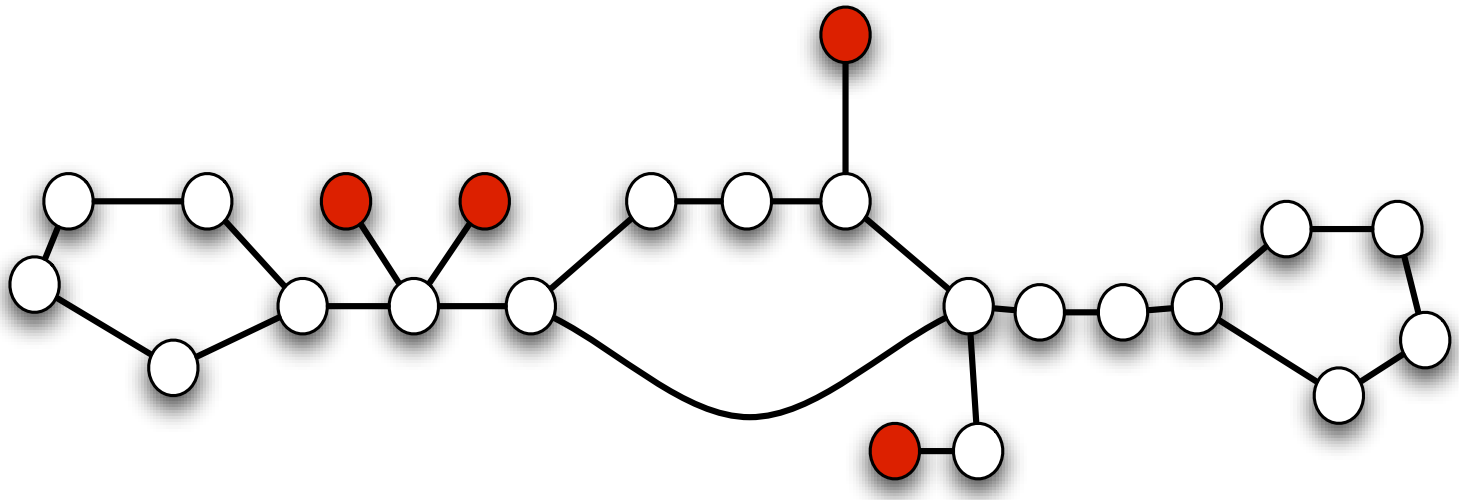
- Vertex labels suppressed so you can see what's going on.
- Leaves (nodes of degree 1) aren't in cycles, so don't affect the cycle structure – remove them!

The cycle structure of a connected graph.



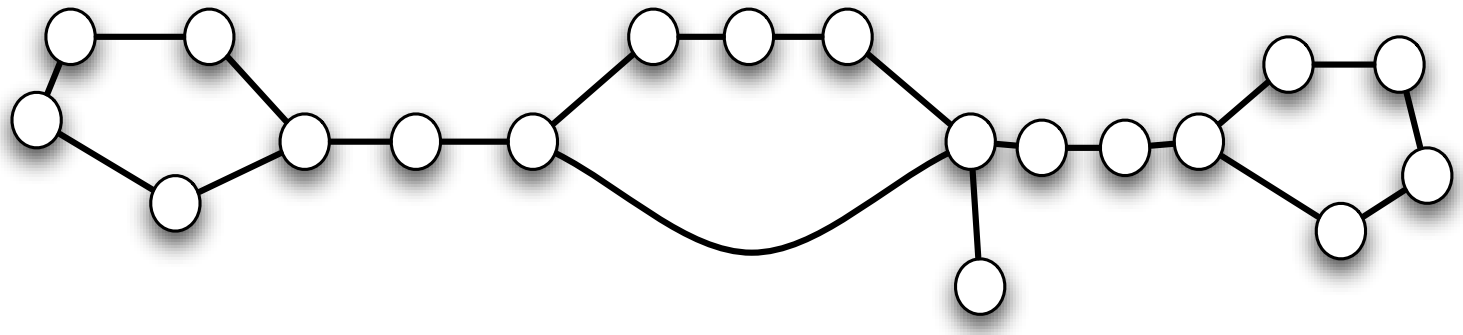
- Vertex labels suppressed so you can see what's going on.
- Leaves (nodes of degree 1) aren't in cycles, so don't affect the cycle structure – remove them!
- More leaves appeared!

The cycle structure of a connected graph.



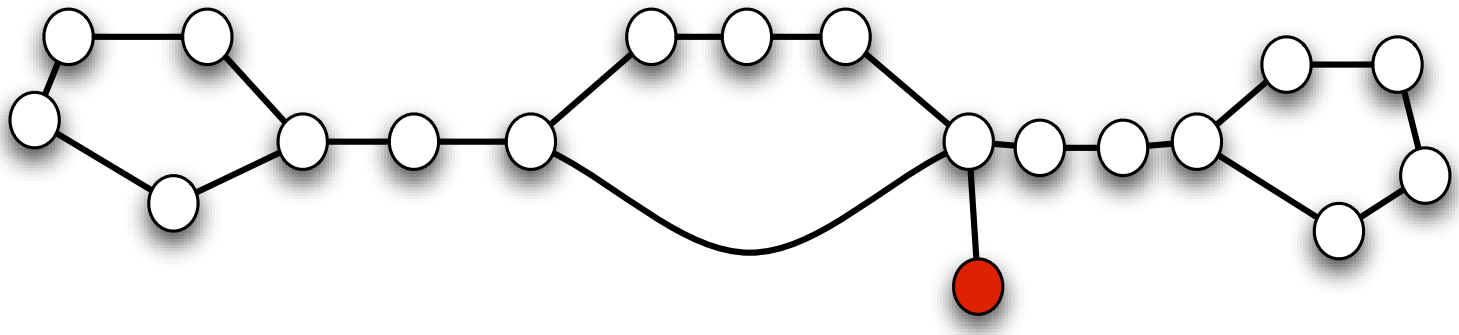
- Vertex labels suppressed so you can see what's going on.
- Leaves (nodes of degree 1) aren't in cycles, so don't affect the cycle structure – remove them!
- More leaves appeared!

The cycle structure of a connected graph.



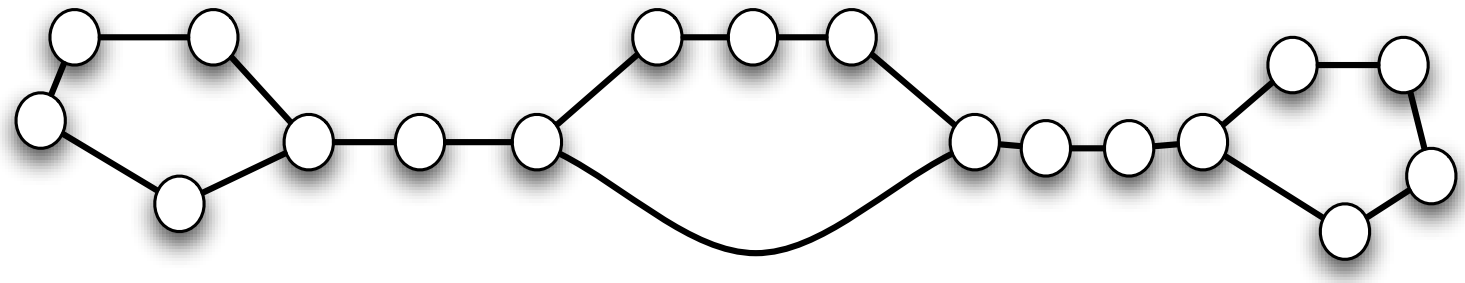
- Vertex labels suppressed so you can see what's going on.
- Leaves (nodes of degree 1) aren't in cycles, so don't affect the cycle structure – remove them!
- More leaves appeared!

The cycle structure of a connected graph.



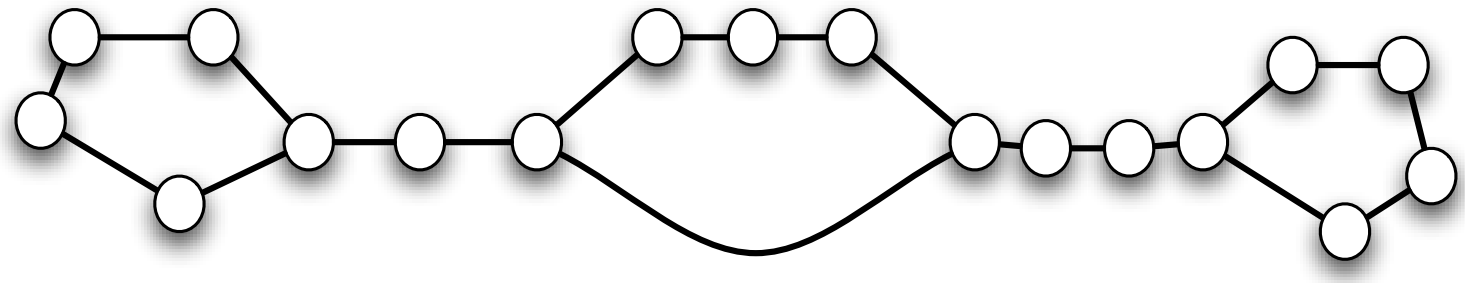
- Vertex labels suppressed so you can see what's going on.
- Leaves (nodes of degree 1) aren't in cycles, so don't affect the cycle structure – remove them!
- More leaves appeared!

The cycle structure of a connected graph.



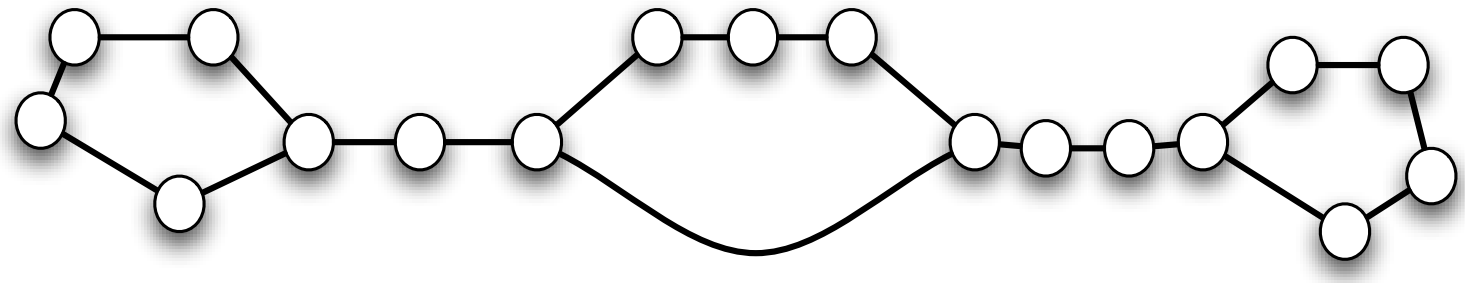
- Vertex labels suppressed so you can see what's going on.
- Leaves (nodes of degree 1) aren't in cycles, so don't affect the cycle structure – remove them!
- More leaves appeared!

The cycle structure of a connected graph.



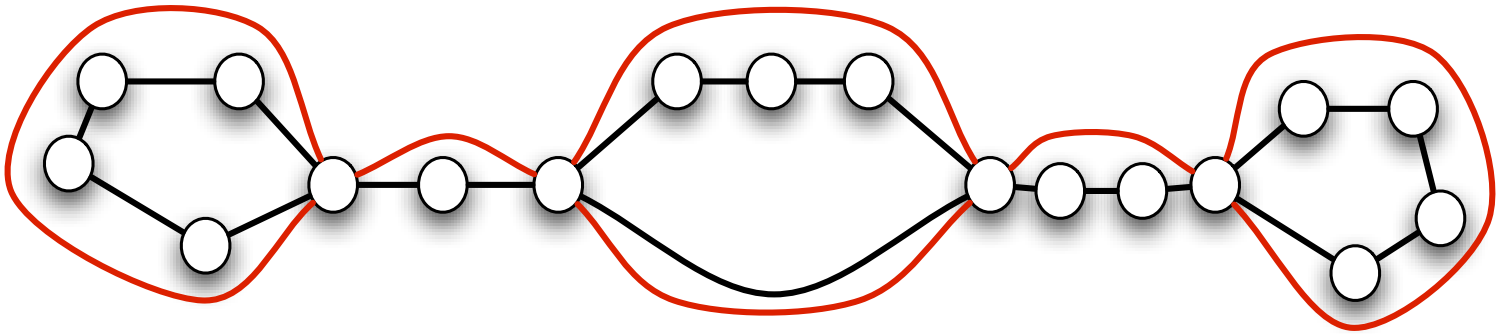
- Vertex labels suppressed so you can see what's going on.
- Leaves (nodes of degree 1) aren't in cycles, so don't affect the cycle structure – remove them!
- What's left when all leaves are gone is called the *core*.

The cycle structure of a connected graph.



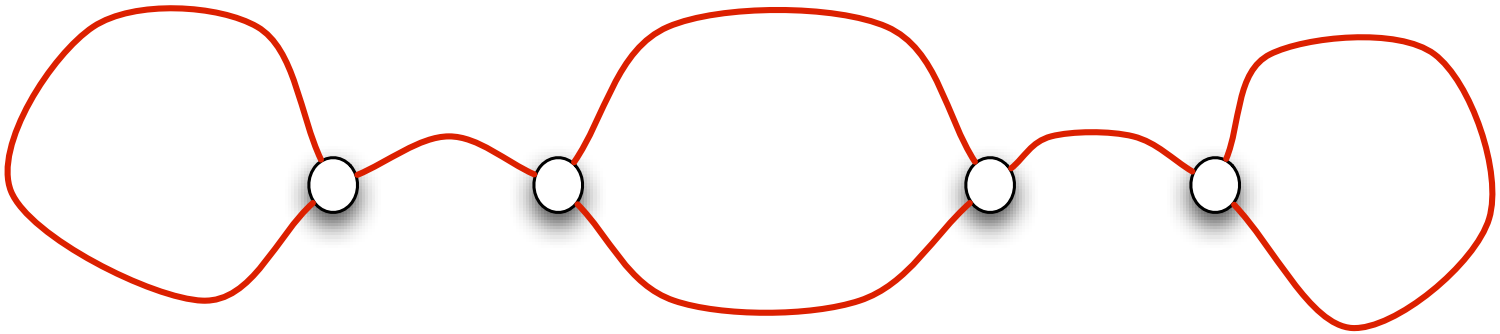
- Vertex labels suppressed so you can see what's going on.
- Leaves (nodes of degree 1) aren't in cycles, so don't affect the cycle structure – remove them!
- What's left when all leaves are gone is called the *core*.
- Vertices of degree 2 in whats left just form paths – suppress them!

The cycle structure of a connected graph.



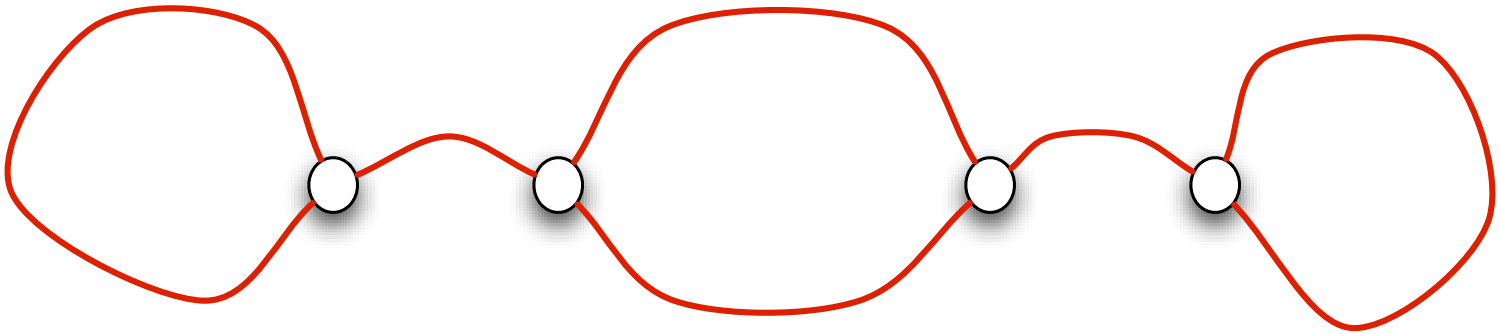
- Vertex labels suppressed so you can see what's going on.
- Leaves (nodes of degree 1) aren't in cycles, so don't affect the cycle structure – remove them!
- What's left when all leaves are gone is called the *core*.
- Vertices of degree 2 in whats left just form paths – suppress them!

The cycle structure of a connected graph.



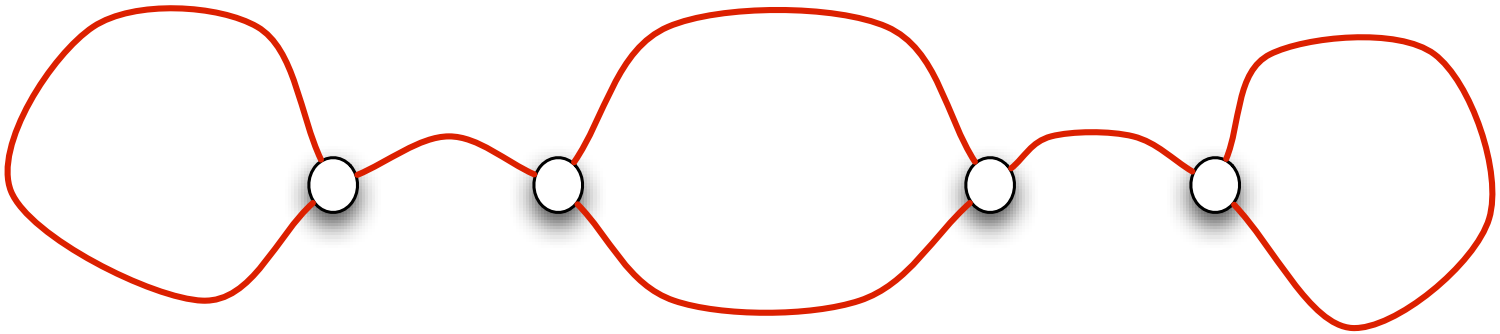
- Vertex labels suppressed so you can see what's going on.
- Leaves (nodes of degree 1) aren't in cycles, so don't affect the cycle structure – remove them!
- What's left when all leaves are gone is called the *core*.
- Vertices of degree 2 in whats left just form paths – suppress them!

The cycle structure of a connected graph.



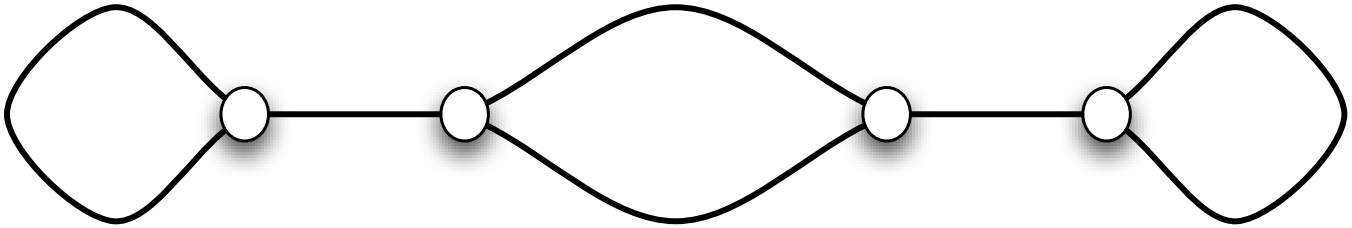
- Vertex labels suppressed so you can see what's going on.
- Leaves (nodes of degree 1) aren't in cycles, so don't affect the cycle structure – remove them!
- What's left when all leaves are gone is called the *core*.
- Vertices of degree 2 in whats left just form paths – suppress them!
- Result after suppressing degree-2 vertices is called the *kernel*.

The cycle structure of a connected graph.



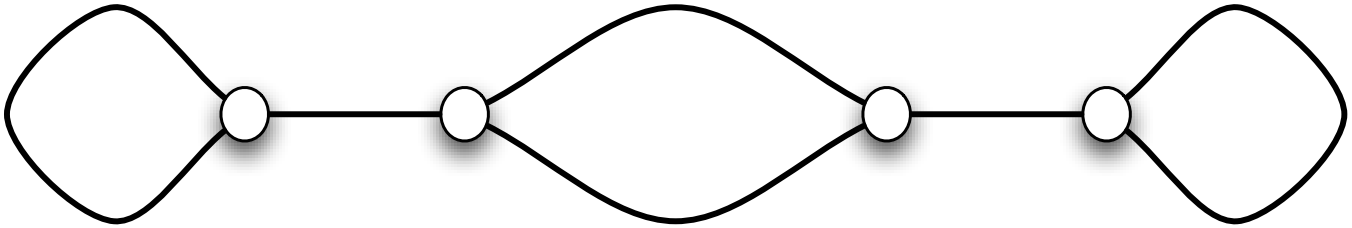
- Vertex labels suppressed so you can see what's going on.
- Leaves (nodes of degree 1) aren't in cycles, so don't affect the cycle structure – remove them!
- What's left when all leaves are gone is called the *core*.
- Vertices of degree 2 in whats left just form paths – suppress them!
- Result after suppressing degree-2 vertices is called the *kernel*.
- Kernel has same surplus as the original graph.

The kernel



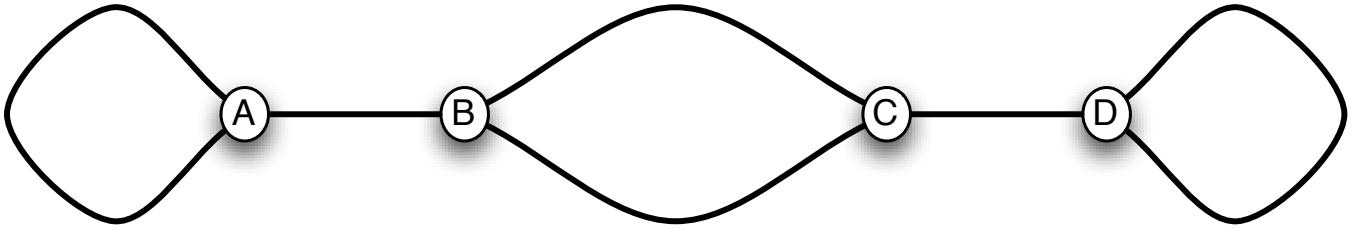
- The kernel is a connected, unlabelled multigraph with minimum degree 3.

The kernel



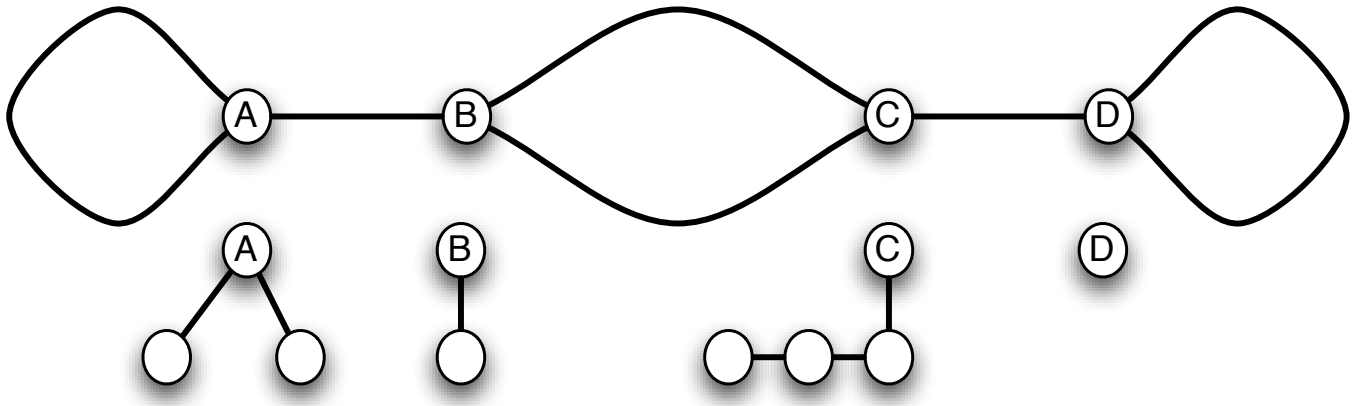
- The kernel is a connected, unlabelled multigraph with minimum degree 3.
- Given the kernel, we can reconstruct the graph by specifying a *marked tree* for each vertex of the kernel, and a *doubly-marked tree* for each edge of the kernel.

The kernel



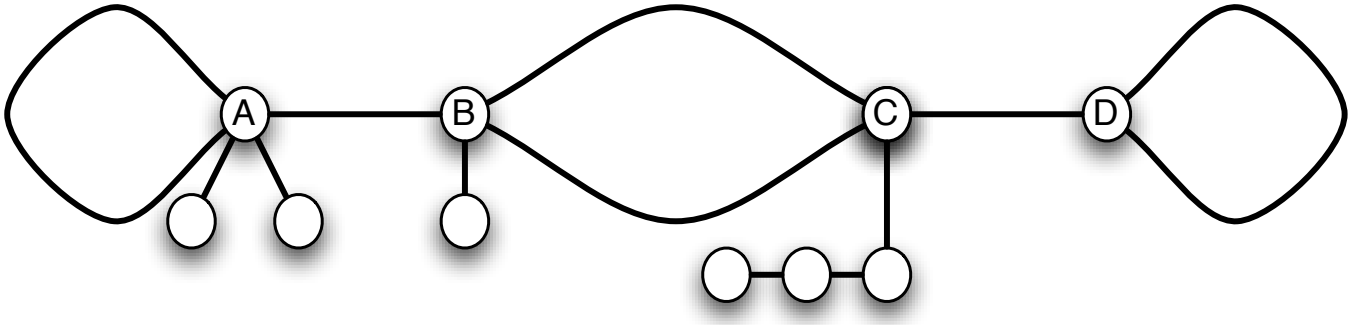
- The kernel is a connected, unlabelled multigraph with minimum degree 3.
- Given the kernel, we can reconstruct the graph by specifying a *marked tree* for each vertex of the kernel, and a *doubly-marked tree* for each edge of the kernel.

The kernel



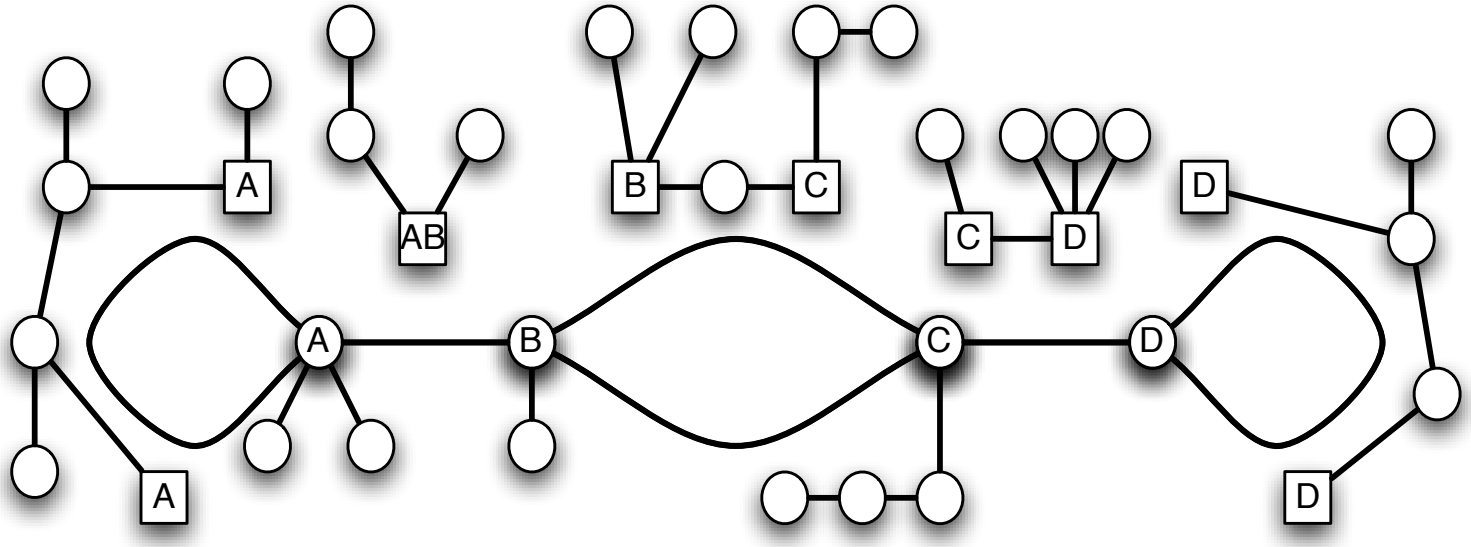
- The kernel is a connected, unlabelled multigraph with minimum degree 3.
- Given the kernel, we can reconstruct the graph by specifying a *marked tree* for each vertex of the kernel, and a *doubly-marked tree* for each edge of the kernel.

The kernel



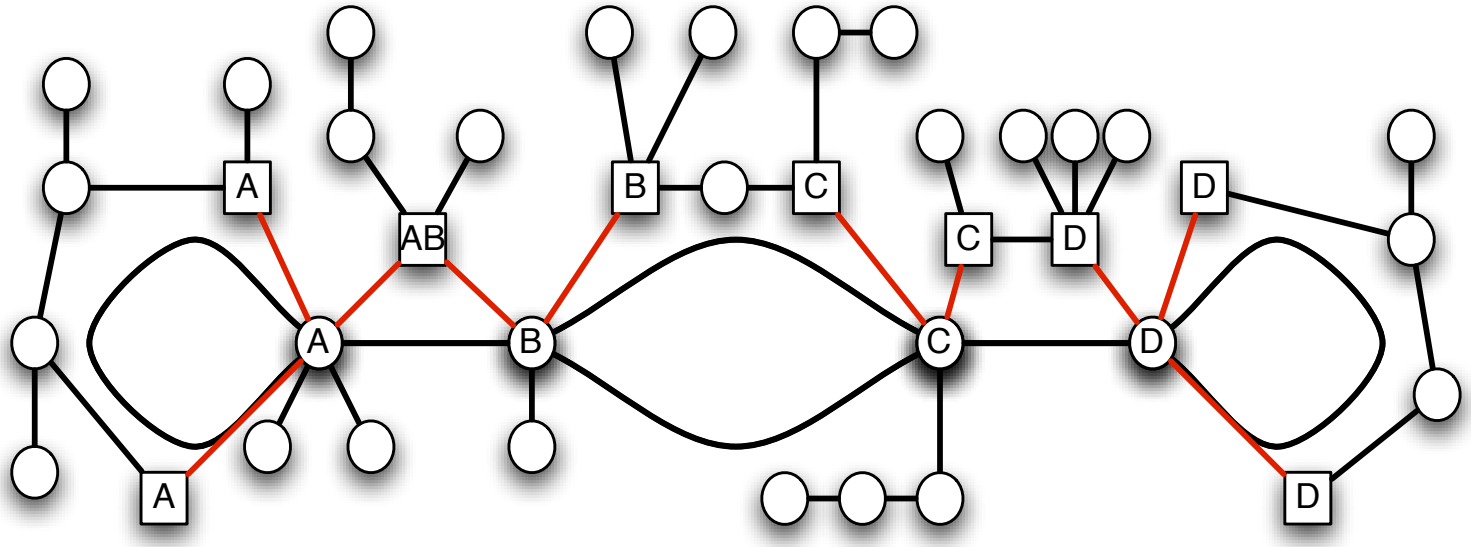
- The kernel is a connected, unlabelled multigraph with minimum degree 3.
- Given the kernel, we can reconstruct the graph by specifying a *marked tree* for each vertex of the kernel, and a *doubly-marked tree* for each edge of the kernel.

The kernel



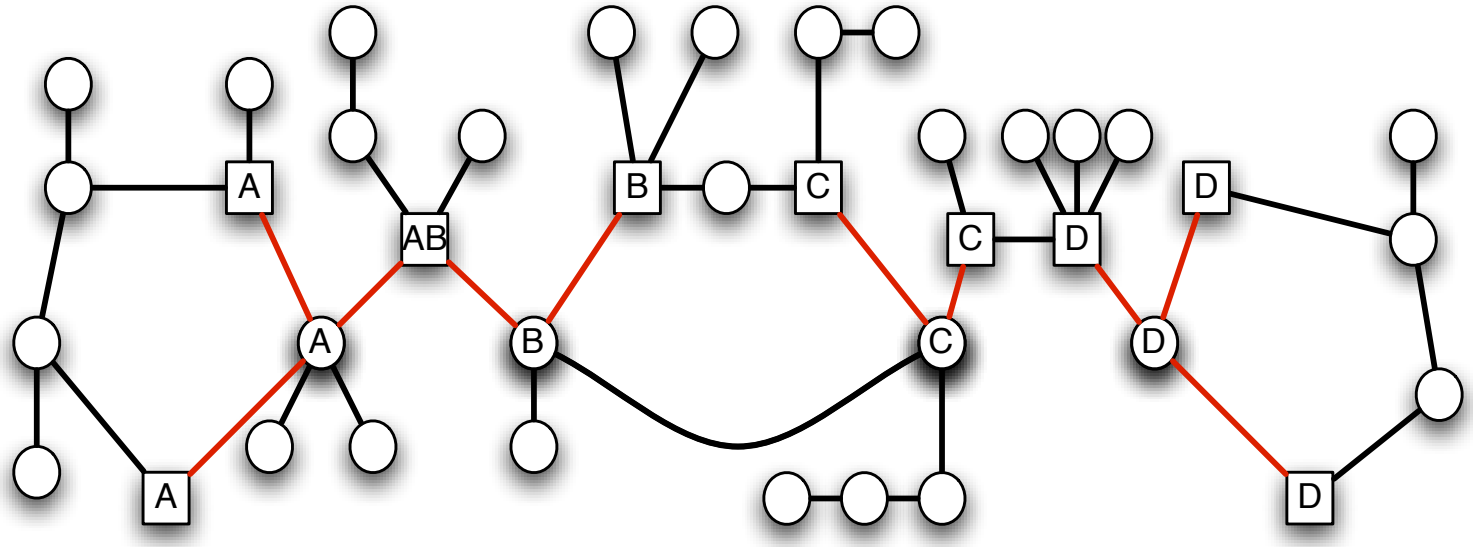
- The kernel is a connected, unlabelled multigraph with minimum degree 3.
- Given the kernel, we can reconstruct the graph by specifying a *marked tree* for each vertex of the kernel, and a *doubly-marked tree* for each edge of the kernel.

The kernel



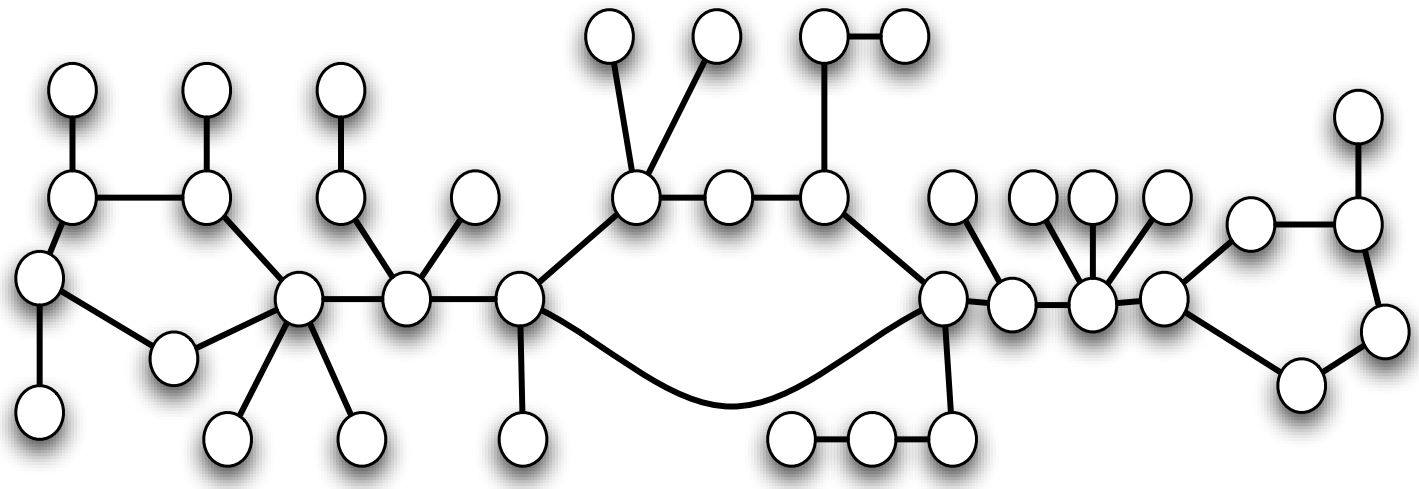
- The kernel is a connected, unlabelled multigraph with minimum degree 3.
- Given the kernel, we can reconstruct the graph by specifying a *marked tree* for each vertex of the kernel, and a *doubly-marked tree* for each edge of the kernel.

The kernel



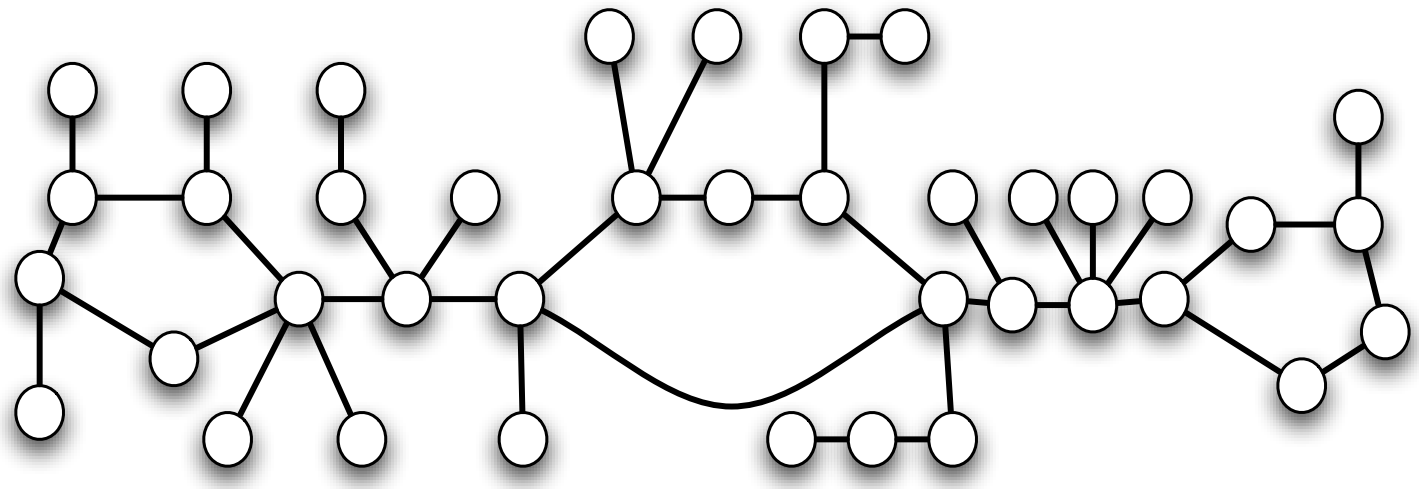
- The kernel is a connected, unlabelled multigraph with minimum degree 3.
- Given the kernel, we can reconstruct the graph by specifying a *marked tree* for each vertex of the kernel, and a *doubly-marked tree* for each edge of the kernel.

The kernel



- The kernel is a connected, unlabelled multigraph with minimum degree 3.
- Given the kernel, we can reconstruct the graph by specifying a *marked tree* for each vertex of the kernel, and a *doubly-marked tree* for each edge of the kernel.

The kernel



- The kernel is a connected, unlabelled multigraph with minimum degree 3.
- Given the kernel, we can reconstruct the graph by specifying a *marked tree* for each vertex of the kernel, and a *doubly-marked tree* for each edge of the kernel.
- Symmetry factors: $1/2$ for loops, $1/k!$ for each k-multiedge.

The kernel of a component

Consider a random connected graph with vertices $1, \ldots, m$ and surplus $s \geq 2$. For s fixed, m large, the structure is as follows.

The kernel of a component

Consider a random connected graph with vertices $1, \ldots, m$ and surplus $s \geq 2$. For s fixed, m large, the structure is as follows.

- With probability $1 - o(1)$ the kernel is 3-regular (all nodes degree three), so has $2(s-1)$ vertices and $3(s-1)$ edges.

The kernel of a component

Consider a random connected graph with vertices $1, \ldots, m$ and surplus $s \geq 2$. For s fixed, m large, the structure is as follows.

- With probability $1 - o(1)$ the kernel is 3-regular (all nodes degree three), so has $2(s-1)$ vertices and $3(s-1)$ edges.
- The probability of any given kernel is the same (up to the symmetry factors mentioned before).

The kernel of a component

Consider a random connected graph with vertices $1, \ldots, m$ and surplus $s \geq 2$. For s fixed, m large, the structure is as follows.

- With probability $1 - o(1)$ the kernel is 3-regular (all nodes degree three), so has $2(s-1)$ vertices and $3(s-1)$ edges.
- The probability of any given kernel is the same (up to the symmetry factors mentioned before).
- "Vertex trees" have $O(1)$ vertices in probability (they disappear in the limit).

The kernel of a component

Consider a random connected graph with vertices $1, \ldots, m$ and surplus $s \geq 2$. For s fixed, m large, the structure is as follows.

- With probability $1 - o(1)$ the kernel is 3-regular (all nodes degree three), so has $2(s-1)$ vertices and $3(s-1)$ edges.
- The probability of any given kernel is the same (up to the symmetry factors mentioned before).
- "Vertex trees" have $O(1)$ vertices in probability (they disappear in the limit).
- List the random "edge trees" as $T_1, \ldots, T_{3(s-1)}$; conditional on their sizes, these trees are *independent uniformly random trees*, so look like rescaled Brownian CRTs in the limit.

The kernel of a component

Consider a random connected graph with vertices $1, \ldots, m$ and surplus $s \geq 2$. For s fixed, m large, the structure is as follows.

- With probability $1 - o(1)$ the kernel is 3-regular (all nodes degree three), so has $2(s-1)$ vertices and $3(s-1)$ edges.
- The probability of any given kernel is the same (up to the symmetry factors mentioned before).
- "Vertex trees" have $O(1)$ vertices in probability (they disappear in the limit).
- List the random "edge trees" as $T_1, \ldots, T_{3(s-1)}$; conditional on their sizes, these trees are *independent uniformly random trees*, so look like rescaled Brownian CRTs in the limit.
- The vector $(|T_1|/m, \ldots, |T_{3(s-1)}|/m)$ has distributional limit Dirichlet$(1/2, \ldots, 1/2)$.

The limiting picture

In the limit, then, a random component can be built by gluing *randomly rescaled Brownian CRTs* along the edges of a random kernel.

How to find a tree in the forest

To find the distribution of MST of a limiting component, use *cycle breaking on the core*.

How to find a tree in the forest

To find the distribution of MST of a limiting component, use *cycle breaking on the core*.
Remark: In the limit, core = kernel with edge lengths, is a random metric space, and edge removal becomes point removal.

How to find a tree in the forest

To find the distribution of MST of a limiting component, use *cycle breaking on the core*.
Remark: In the limit, core = kernel with edge lengths, is a random metric space, and edge removal becomes point removal.
As before, random connected graph with vertices $1, \ldots, m$ and surplus $s \geq 2$.

How to find a tree in the forest

To find the distribution of MST of a limiting component, use *cycle breaking on the core*.
Remark: In the limit, core = kernel with edge lengths, is a random metric space, and edge removal becomes point removal.
As before, random connected graph with vertices $1, \ldots, m$ and surplus $s \geq 2$.

- Write C for the number of vertices in the core – then $C = O(\sqrt{m})$ in expectation and $C/\sqrt{m}$ has a non-degenerate limit.

How to find a tree in the forest

To find the distribution of MST of a limiting component, use *cycle breaking on the core*.
Remark: In the limit, core = kernel with edge lengths, is a random metric space, and edge removal becomes point removal.
As before, random connected graph with vertices $1, \ldots, m$ and surplus $s \geq 2$.

- Write C for the number of vertices in the core – then $C = O(\sqrt{m})$ in expectation and $C/\sqrt{m}$ has a non-degenerate limit.
- Write $P_1, \ldots, P_{3(s-1)}$ for the core paths corresponding to kernel edges – then $(|P_1|/C, \ldots, |P_{3(s-1)}/C)$ has distributional limit Dirichlet$(1, \ldots, 1)$.

How to find a tree in the forest

To find the distribution of MST of a limiting component, use *cycle breaking on the core*.
Remark: In the limit, core = kernel with edge lengths, is a random metric space, and edge removal becomes point removal.
As before, random connected graph with vertices $1, \ldots, m$ and surplus $s \geq 2$.

- Write C for the number of vertices in the core – then $C = O(\sqrt{m})$ in expectation and $C/\sqrt{m}$ has a non-degenerate limit.
- Write $P_1, \ldots, P_{3(s-1)}$ for the core paths corresponding to kernel edges – then $(|P_1|/C, \ldots, |P_{3(s-1)}/C)$ has distributional limit Dirichlet$(1, \ldots, 1)$.
- In other words, core path lengths are just given by splittings of an interval by uniforms.

Cycle breaking on the core

Choose the largest weight edge in the core, try to remove it.

Cycle breaking on the core

Choose the largest weight edge in the core, try to remove it.
This edge is equally likely to be any edge of the core.

Cycle breaking on the core

Choose the largest weight edge in the core, try to remove it.
This edge is equally likely to be any edge of the core.
In the limit, edge lengths shrink to zero and we're choosing a random point on the core.

Cycle breaking on the core

Choose the largest weight edge in the core, try to remove it.
This edge is equally likely to be any edge of the core.
In the limit, edge lengths shrink to zero and we're choosing a random point on the core.
This may or may not break a cycle.

Cycle breaking on the core

Choose the largest weight edge in the core, try to remove it.
This edge is equally likely to be any edge of the core.
In the limit, edge lengths shrink to zero and we're choosing a random point on the core.
This may or may not break a cycle.
But it certainly breaks an edge tree in two.

Cycle breaking on the core

Choose the largest weight edge in the core, try to remove it.
This edge is equally likely to be any edge of the core.
In the limit, edge lengths shrink to zero and we're choosing a random point on the core.
This may or may not break a cycle.
But it certainly breaks an edge tree in two.
Resulting trees are still (rescaled) Brownian CRTs!

Cycle breaking on the core

Choose the largest weight edge in the core, try to remove it.
This edge is equally likely to be any edge of the core.
In the limit, edge lengths shrink to zero and we're choosing a random point on the core.
This may or may not break a cycle.
But it certainly breaks an edge tree in two.
Resulting trees are still (rescaled) Brownian CRTs!
Vector of edge tree masses still Dirichlet$(1/2, \ldots, 1/2)$ with length increased by one.

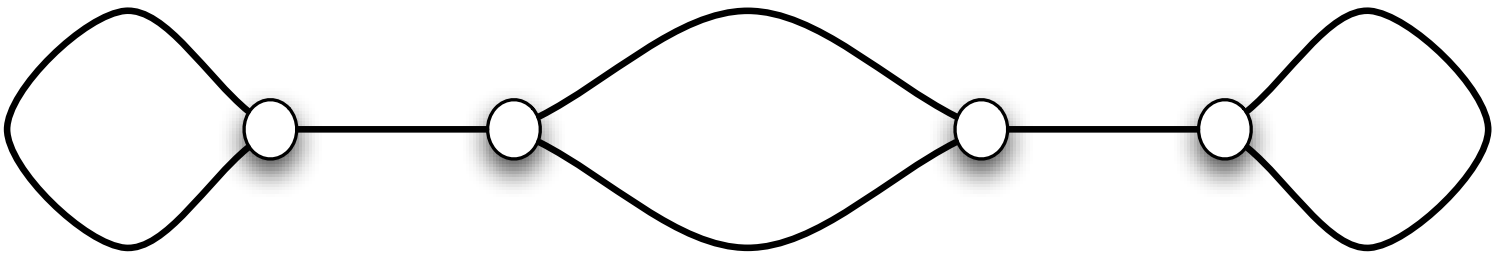
Cycle breaking on the core

Choose the largest weight edge in the core, try to remove it.
This edge is equally likely to be any edge of the core.
In the limit, edge lengths shrink to zero and we're choosing a random point on the core.
This may or may not break a cycle.
But it certainly breaks an edge tree in two.
Resulting trees are still (rescaled) Brownian CRTs!
Vector of edge tree masses still Dirichlet$(1/2, \ldots, 1/2)$ with length increased by one.
Vector of core lengths still Dirichlet$(1, \ldots, 1)$ with length increased by one.

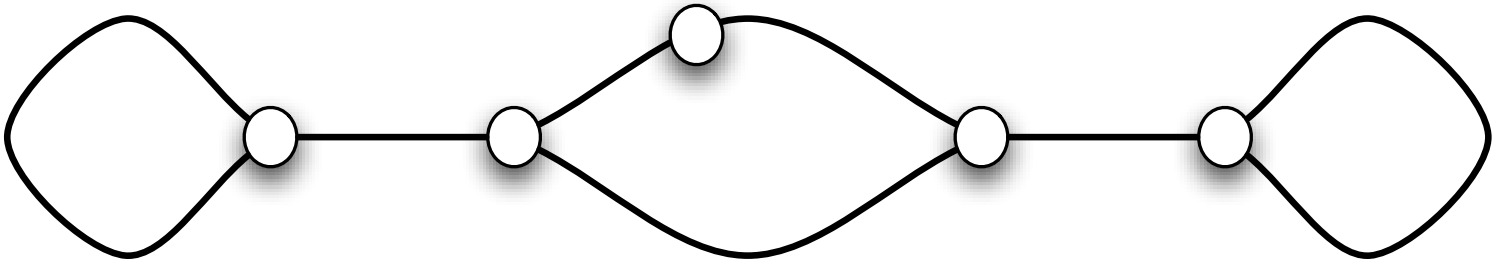
Cycle breaking on the core

Choose the largest weight edge in the core, try to remove it.
This edge is equally likely to be any edge of the core.
In the limit, edge lengths shrink to zero and we're choosing a random point on the core.
This may or may not break a cycle.
But it certainly breaks an edge tree in two.
Resulting trees are still (rescaled) Brownian CRTs!
Vector of edge tree masses still Dirichlet$(1/2, \ldots, 1/2)$ with length increased by one.
Vector of core lengths still Dirichlet$(1, \ldots, 1)$ with length increased by one.
Repeat until all cycles are broken.

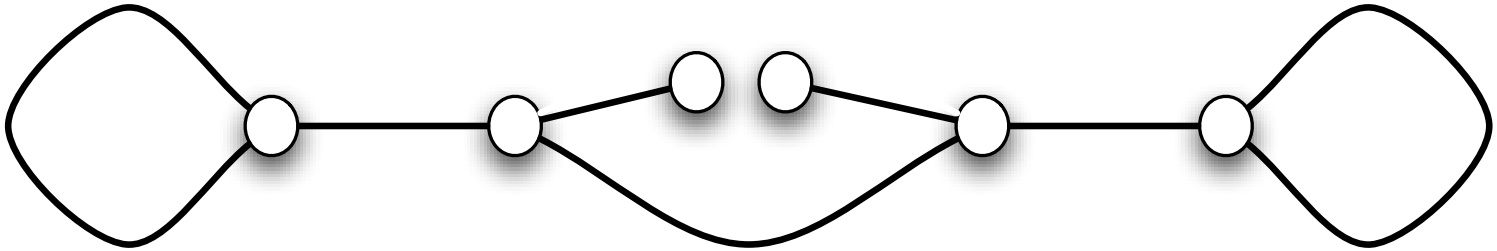
Cycle breaking on the core



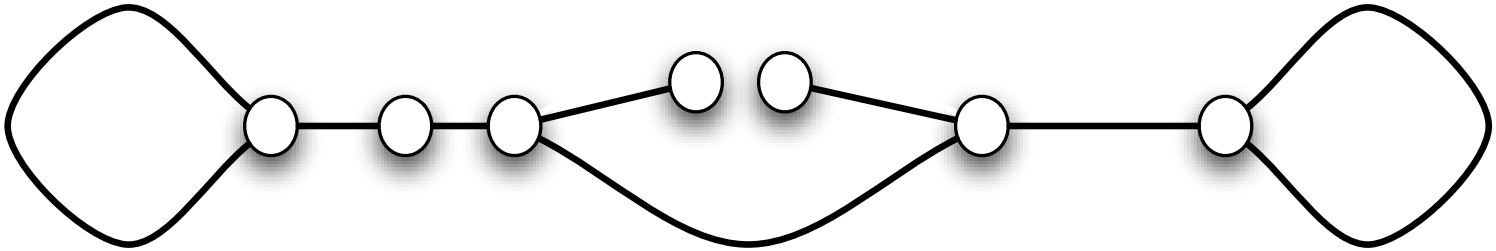
Cycle breaking on the core



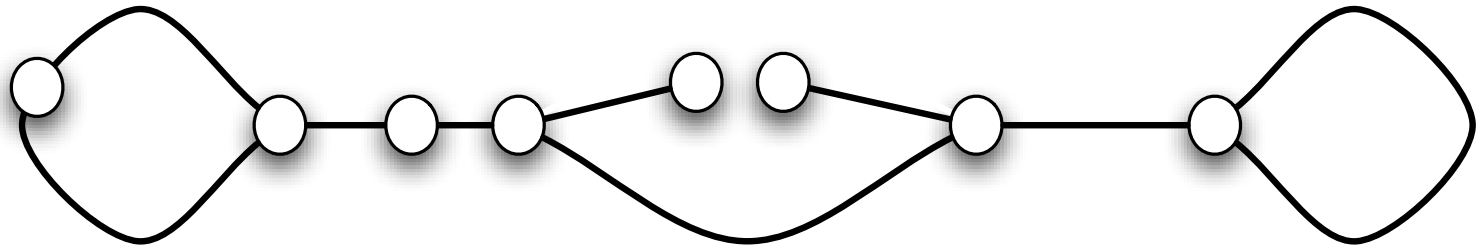
Cycle breaking on the core



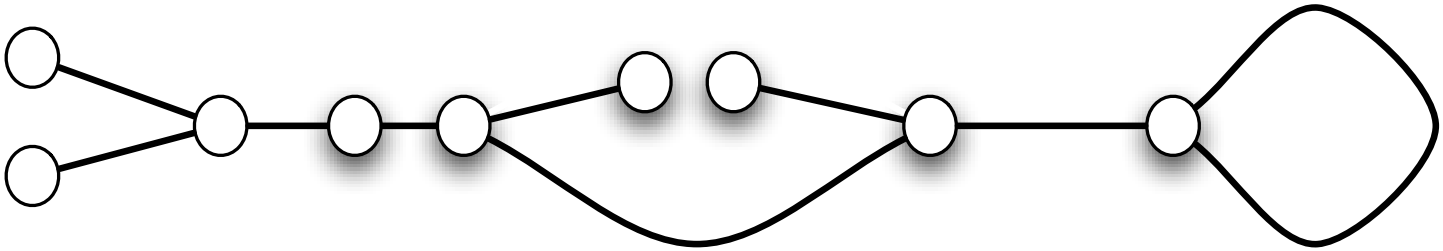
Cycle breaking on the core



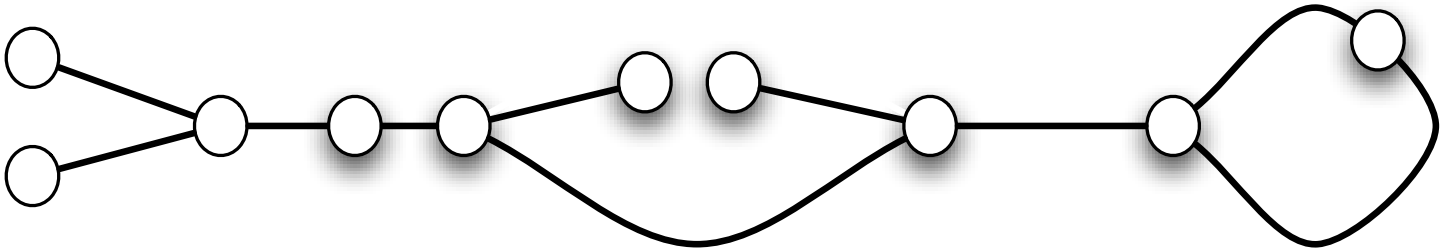
Cycle breaking on the core



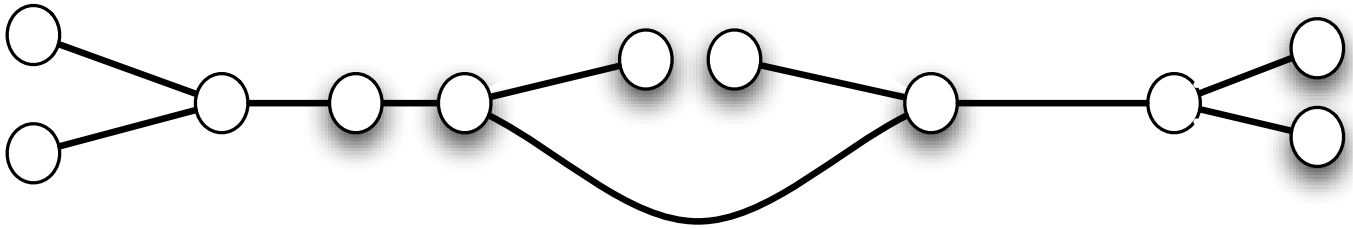
Cycle breaking on the core



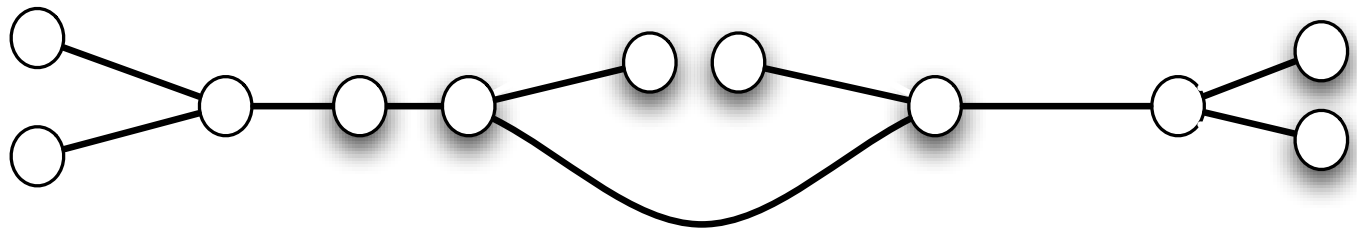
Cycle breaking on the core



Cycle breaking on the core

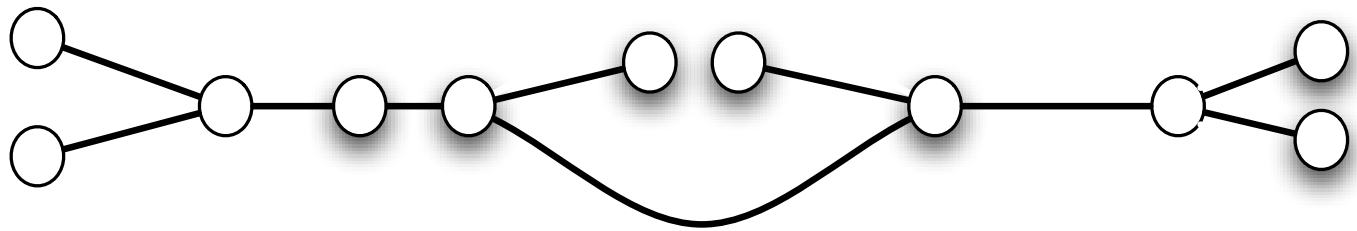


Cycle breaking on the core



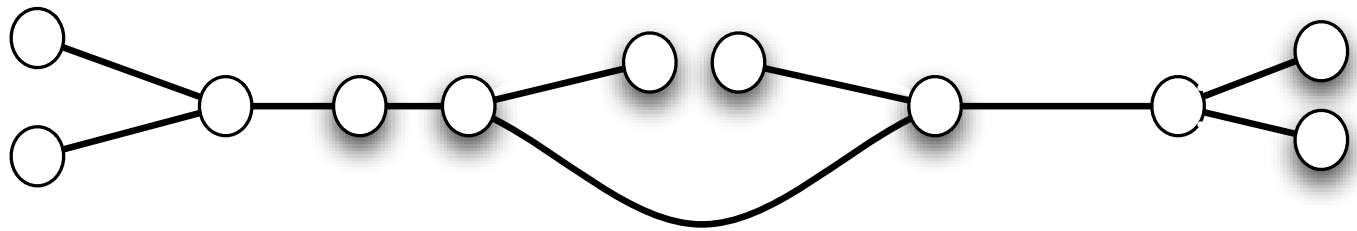
Result: MST of a random connected graph with a fixed surplus s, can be built (in the limit) by gluing randomly rescaled CRTs along the edges of a random discrete tree

Cycle breaking on the core



Result: MST of a random connected graph with a fixed surplus s, can be built (in the limit) by gluing randomly rescaled CRTs along the edges of a random discrete tree whose internal nodes all have degrees 2 or 3.

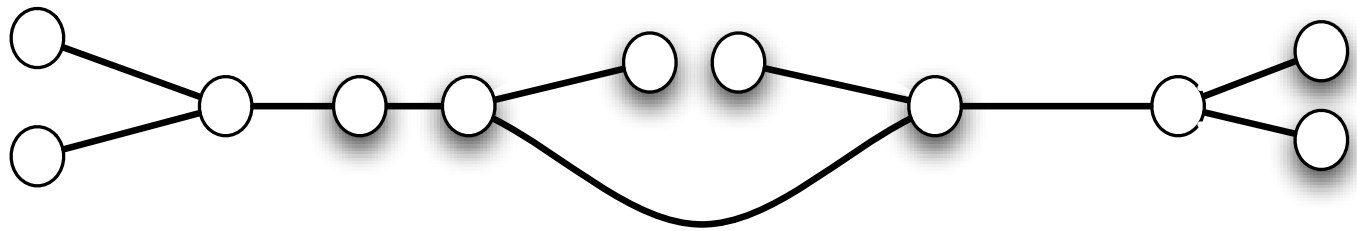
Cycle breaking on the core



Result: MST of a random connected graph with a fixed surplus s, can be built (in the limit) by gluing randomly rescaled CRTs along the edges of a random discrete tree whose internal nodes all have degrees 2 or 3.

We have not found a nice formula for the distribution of this discrete tree.

Cycle breaking on the core



Result: MST of a random connected graph with a fixed surplus s, can be built (in the limit) by gluing randomly rescaled CRTs along the edges of a random discrete tree whose internal nodes all have degrees 2 or 3.

We have not found a nice formula for the distribution of this discrete tree.

In the $s \to \infty$ limit, it must have height of order $s^{1/3}$, because in the $s \to \infty$ limit it encodes the structure of the MST of K_n, and this is known to have order $n^{1/3}$.

Proof idea

Theorem

As $n \to \infty$, $T_{n,1/n} \xrightarrow{\mathrm{d}} \mathcal{T}$, *for some sequence* $\mathcal{T} = (\mathcal{T}_i, i \geq 1)$ *of random metric spaces.* (*Component−wise GHP, plus tail decay estimates.*)

Idea: The largest component $\mathcal{T}_1$ will tell us the structure of $\mathcal{M}$.

Proof idea

Theorem

As $n \to \infty$, $T_{n,1/n} \xrightarrow{\mathrm{d}} \mathcal{T}$, *for some sequence* $\mathcal{T} = (\mathcal{T}_i, i \geq 1)$ *of random metric spaces.* (*Component−wise GHP, plus tail decay estimates.*)

Idea: The largest component $\mathcal{T}_1$ will tell us the structure of $\mathcal{M}$.

- Not true at $p = 1/n$;

Proof idea

Theorem

As $n \to \infty$, $T_{n,1/n} \xrightarrow{\mathrm{d}} \mathcal{T}$, *for some sequence* $\mathcal{T} = (\mathcal{T}_i, i \geq 1)$ *of random metric spaces.* (*Component−wise GHP, plus tail decay estimates.*)

Idea: The largest component $\mathcal{T}_1$ will tell us the structure of $\mathcal{M}$.

- Not true at $p = 1/n$; true at $p = 2/n$ (but lose control of $G_{n,p}$);

Proof idea

Theorem

As $n \to \infty$, $T_{n,1/n} \xrightarrow{\mathrm{d}} \mathcal{T}$, *for some sequence* $\mathcal{T} = (\mathcal{T}_i, i \geq 1)$ *of random metric spaces.* (*Component−wise GHP, plus tail decay estimates.*)

Idea: The largest component $\mathcal{T}_1$ will tell us the structure of $\mathcal{M}$.

- Not true at $p = 1/n$; true at $p = 2/n$ (but lose control of $G_{n,p}$); *basically* true at $p = 1/n + \lambda/n^{4/3}$, for λ large.

Proof idea

Theorem

As $n \to \infty$, $T_n^\lambda \xrightarrow{\mathrm{d}} \mathcal{T}^\lambda$, *for some sequence* $\mathcal{T}^\lambda = (\mathcal{T}_i^\lambda, i \geq 1)$ *of random metric spaces.* (*Component−wise GHP, plus tail decay estimates.*)

Idea: The largest component $\mathcal{T}_1$ will tell us the structure of $\mathcal{M}$.

- Not true at $p = 1/n$; true at $p = 2/n$ (but lose control of $G_{n,p}$); *basically* true at $p = 1/n + \lambda/n^{4/3}$, for λ large.

Proof idea

Theorem

As $n \to \infty$, $T_n^\lambda \xrightarrow{\mathrm{d}} \mathcal{T}^\lambda$, *for some sequence* $\mathcal{T}^\lambda = (\mathcal{T}_i^\lambda, i \geq 1)$ *of random metric spaces.* (*Component−wise GHP, plus tail decay estimates.*)

Idea: The largest component $\mathcal{T}_1$ will tell us the structure of $\mathcal{M}$.

- Not true at $p = 1/n$; true at $p = 2/n$ (but lose control of $G_{n,p}$); *basically* true at $p = 1/n + \lambda/n^{4/3}$, for λ large.
- Steps:

Proof idea

Theorem

As $n \to \infty$, $T_n^\lambda \xrightarrow{\mathrm{d}} \mathcal{T}^\lambda$, *for some sequence* $\mathcal{T}^\lambda = (\mathcal{T}_i^\lambda, i \geq 1)$ *of random metric spaces.* (*Component−wise GHP, plus tail decay estimates.*)

Idea: The largest component $\mathcal{T}_1$ will tell us the structure of $\mathcal{M}$.

- Not true at $p = 1/n$; true at $p = 2/n$ (but lose control of $G_{n,p}$); *basically* true at $p = 1/n + \lambda/n^{4/3}$, for λ large.
- Steps:
  (1) Show that the largest diameter of any component except $\mathcal{T}_1^\lambda$ tends to 0 in probability as $\lambda \to \infty$ (roughly speaking, this is due to a duality principle).

Proof idea

Theorem

As $n \to \infty$, $T_n^\lambda \xrightarrow{\mathrm{d}} \mathcal{T}^\lambda$, *for some sequence* $\mathcal{T}^\lambda = (\mathcal{T}_i^\lambda, i \geq 1)$ *of random metric spaces.* (*Component−wise GHP, plus tail decay estimates.*)

Idea: The largest component $\mathcal{T}_1$ will tell us the structure of $\mathcal{M}$.

- Not true at $p = 1/n$; true at $p = 2/n$ (but lose control of $G_{n,p}$); *basically* true at $p = 1/n + \lambda/n^{4/3}$, for λ large.
- Steps:
  (1) Show that the largest diameter of any component except $\mathcal{T}_1^\lambda$ tends to 0 in probability as $\lambda \to \infty$ (roughly speaking, this is due to a duality principle).
  (2) Show that the mass in $(\mathcal{T}_i^\lambda, i \geq 2)$ attaches roughly uniformly to $\mathcal{T}_1^\lambda$, so the mass measure of $\mathcal{T}_1^\lambda$ is already a good approximation for the limit (Glivenko-Cantelli + additional arguments).

Proof idea

Theorem

As $n \to \infty$, $T_n^\lambda \xrightarrow{\mathrm{d}} \mathcal{T}^\lambda$, for some sequence $\mathcal{T}^\lambda = (\mathcal{T}_i^\lambda, i \geq 1)$ of random metric spaces. (*Component−wise GHP, plus tail decay estimates.*)

Idea: The largest component $\mathcal{T}_1$ will tell us the structure of $\mathcal{M}$.

- Not true at $p = 1/n$; true at $p = 2/n$ (but lose control of $G_{n,p}$); *basically* true at $p = 1/n + \lambda/n^{4/3}$, for λ large.
- Steps:
  (1) Show that the largest diameter of any component except $\mathcal{T}_1^\lambda$ tends to 0 in probability as $\lambda \to \infty$ (roughly speaking, this is due to a duality principle).
  (2) Show that the mass in $(\mathcal{T}_i^\lambda, i \geq 2)$ attaches roughly uniformly to $\mathcal{T}_1^\lambda$, so the mass measure of $\mathcal{T}_1^\lambda$ is already a good approximation for the limit (Glivenko-Cantelli + additional arguments).

The principle of accompanying laws then yields that
$\lim_{n\to\infty} T_n = \lim_{n\to\infty} \lim_{\lambda\to\infty} T_{n,1}^\lambda = \lim_{\lambda\to\infty} T_1^\lambda = \mathcal{M}$,
for some random metric space $\mathcal{M}$. □

The structure of the limit $\mathcal{M}$.

1. $\mathcal{M}$ is a random $\mathbb{R}$-tree (geodesic space, unique geodesic between any two points).

The structure of the limit $\mathcal{M}$.

1. $\mathcal{M}$ is a random $\mathbb{R}$-tree (geodesic space, unique geodesic between any two points).
2. It is binary (all points degree ≤ 3); its mass measure is concentrated its leaves.

The structure of the limit $\mathcal{M}$.

1. $\mathcal{M}$ is a random $\mathbb{R}$-tree (geodesic space, unique geodesic between any two points).
2. It is binary (all points degree ≤ 3); its mass measure is concentrated its leaves.
3. $\mathcal{M}$ has Minkowski dimension 3, and so:

The structure of the limit $\mathcal{M}$.

1. $\mathcal{M}$ is a random $\mathbb{R}$-tree (geodesic space, unique geodesic between any two points).
2. It is binary (all points degree ≤ 3); its mass measure is concentrated its leaves.
3. $\mathcal{M}$ has Minkowski dimension 3, and so:
4. the law of $\mathcal{M}$ is mutually singular with that of the Brownian CRT (which has Minkowski dimension 2).

The structure of the limit $\mathcal{M}$.

$\mathcal{M}$ has box-counting dimension 3.

The structure of the limit $\mathcal{M}$.

$\mathcal{M}$ has box-counting dimension 3.

$\star$ Key idea: $\mathcal{T}_1^\lambda$ is built out of $\approx \lambda^3$ tiny Brownian CRT's,

The structure of the limit $\mathcal{M}$.

$\mathcal{M}$ has box-counting dimension 3.

$\star$ Key idea: $\mathcal{T}_1^\lambda$ is built out of $\approx \lambda^3$ tiny Brownian CRT's, glued onto the edges of a certain random combinatorial tree.

The structure of the limit $\mathcal{M}$.

$\mathcal{M}$ has box-counting dimension 3.

$\star$ Key idea: $\mathcal{T}_1^\lambda$ is built out of $\approx \lambda^3$ tiny Brownian CRT's, glued onto the edges of a certain random combinatorial tree.

$\star$ This is because largest comp. of $G_{n,1/n+\lambda/n^{4/3}}$ has $\Theta_p(\lambda^3)$ core paths.

The structure of the limit $\mathcal{M}$.

$\mathcal{M}$ has box-counting dimension 3.

$\star$ Key idea: $\mathcal{T}_1^\lambda$ is built out of $\approx \lambda^3$ tiny Brownian CRT's, glued onto the edges of a certain random combinatorial tree.

$\star$ This is because largest comp. of $G_{n,1/n+\lambda/n^{4/3}}$ has $\Theta_p(\lambda^3)$ core paths.

$\star$ Each core path is "really" a path between two uniform vertices in a CRT of mass $\Theta_p(\lambda^{-2})$.

The structure of the limit $\mathcal{M}$.

$\mathcal{M}$ has box-counting dimension 3.

$\star$ Key idea: $\mathcal{T}_1^\lambda$ is built out of $\approx \lambda^3$ tiny Brownian CRT's, glued onto the edges of a certain random combinatorial tree.

$\star$ This is because largest comp. of $G_{n,1/n+\lambda/n^{4/3}}$ has $\Theta_p(\lambda^3)$ core paths.

$\star$ Each core path is "really" a path between two uniform vertices in a CRT of mass $\Theta_p(\lambda^{-2})$.

$\star$ Brownian scaling then implies that core paths are typically of length $\Theta_p(\lambda^{-1})$.

The structure of the limit $\mathcal{M}$.

$\mathcal{M}$ has box-counting dimension 3.

$\star$ Key idea: $\mathcal{T}_1^\lambda$ is built out of $\approx \lambda^3$ tiny Brownian CRT's, glued onto the edges of a certain random combinatorial tree.

$\star$ This is because largest comp. of $G_{n,1/n+\lambda/n^{4/3}}$ has $\Theta_p(\lambda^3)$ core paths.

$\star$ Each core path is "really" a path between two uniform vertices in a CRT of mass $\Theta_p(\lambda^{-2})$.

$\star$ Brownian scaling then implies that core paths are typically of length $\Theta_p(\lambda^{-1})$.

$\star$ The midpoints of core paths then forms a family of $\approx \lambda^3$ points at pairwise distance $\approx \lambda^{-1}$, which yields that the Minkowski dimension of the limit is ≥ 3.

The structure of the limit $\mathcal{M}$.

$\mathcal{M}$ has box-counting dimension 3.

$\star$ Key idea: $\mathcal{T}_1^\lambda$ is built out of $\approx \lambda^3$ tiny Brownian CRT's, glued onto the edges of a certain random combinatorial tree.

$\star$ This is because largest comp. of $G_{n,1/n+\lambda/n^{4/3}}$ has $\Theta_p(\lambda^3)$ core paths.

$\star$ Each core path is "really" a path between two uniform vertices in a CRT of mass $\Theta_p(\lambda^{-2})$.

$\star$ Brownian scaling then implies that core paths are typically of length $\Theta_p(\lambda^{-1})$.

$\star$ The midpoints of core paths then forms a family of $\approx \lambda^3$ points at pairwise distance $\approx \lambda^{-1}$, which yields that the Minkowski dimension of the limit is ≥ 3.

$\star$ On the other hand, the proportion of points not covered by balls of radius $C\lambda^{-1}$ whp decays exponentially in C; together with estimates on the GH distance between $\mathcal{T}_\lambda$ and $\mathcal{M}$, this yields that the Minkowski dimension of the limit is ≤ 3.