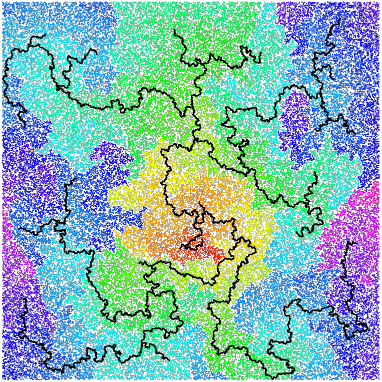
The critical random graph 2



L. Addario-Berry

Random Structures and Dynamics

Oxford

April 11-14, 2011

Parts are joint with Nicolas Broutin, Luc Devroye, Christina Goldschmidt, Svante Janson, Grégory Miermont

Percolation on the complete graph

- Complete graph K_n: vertices $1, \dots, n$ 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 $1, \dots, n$ 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.

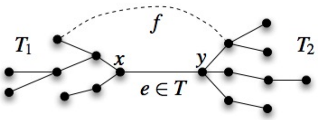
Percolation on the complete graph

- 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.

The tree process and the graph process

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

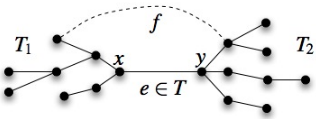
$\forall f$ from T_1 to T_2, $w_f > w_e$.



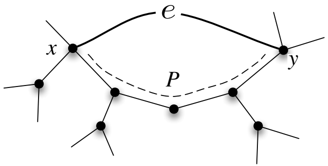
The tree process and the graph process

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, $w_f > w_e$.



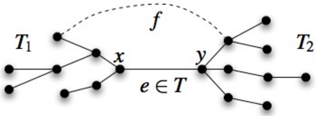
Every edge e of P satisfies $w_e < w_f$



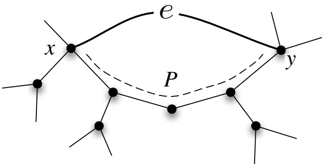
The tree process and the graph process

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 $w_e < w_f$

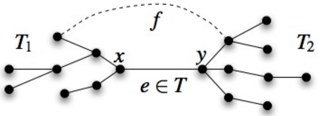


Kruskal's algorithm adds an edge $e = x, y$ of weight p if and only if x and y are in distinct components of $G_{n,p}$.

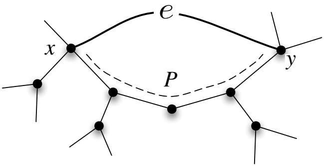
The tree process and the graph process

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).

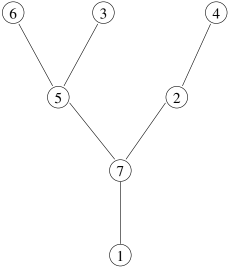
Erdős & Renyi, 1960

- 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 $1/n$.

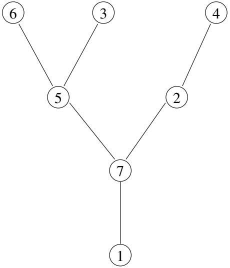
A reminder about trees.

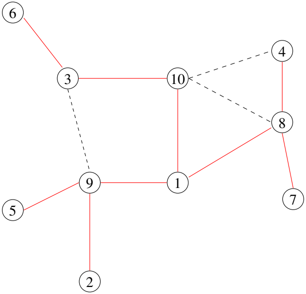
A tree is a connected graph with no cycles. A tree always has one more vertex than it has edges.



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

Theorem (Aldous, 1997)

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

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

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

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

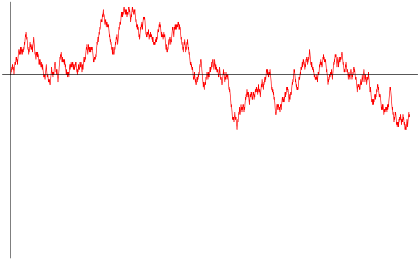
- The second is of finite-dimensional distributions.

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

What are the limit sequences?

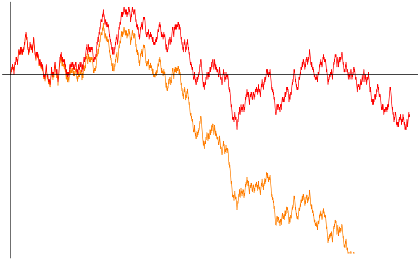
Convergence of component sizes of $G_{n,1/n}$.

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



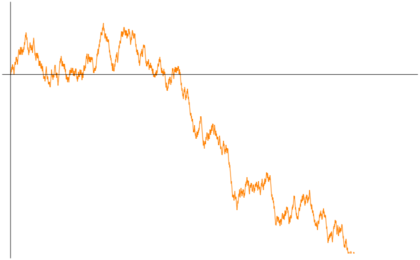
Convergence of component sizes of $G_{n,1/n}$.

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



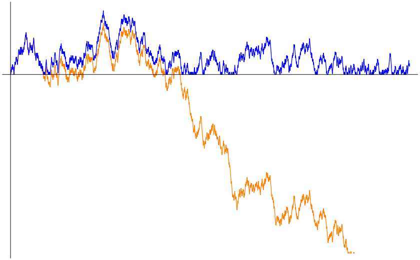
Convergence of component sizes of $G_{n,1/n}$.

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



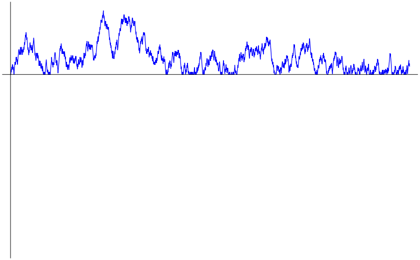
Convergence of component sizes of $G_{n,1/n}$.

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



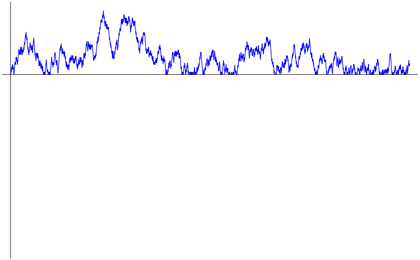
Convergence of component sizes of $G_{n,1/n}$.

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



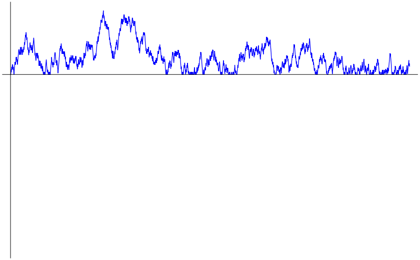
Convergence of component sizes of $G_{n,1/n}$.

- 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).



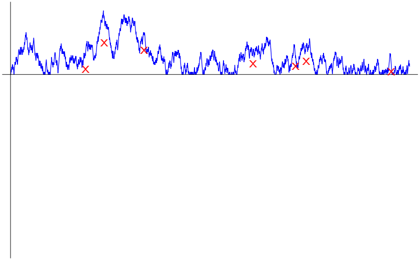
Convergence of component sizes of $G_{n,1/n}$.

- 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^*.



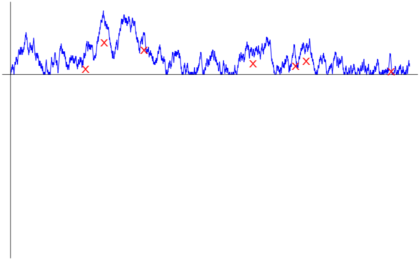
Convergence of component sizes of $G_{n,1/n}$.

- 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:

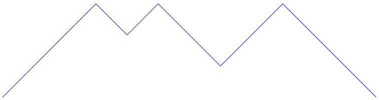
- The *excursion* is encoding a certain spanning tree of the component.
- The *marks* are telling you where to put the extra edges to recover the component from its spanning tree.

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.

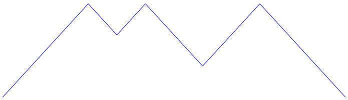
Given any continuous excursion, we can define an associated tree (metric space), by gluing together points of the excursion that (a) have the same height and (b) can see each other.



[Picture by Christina Goldschmidt.]

Trees coded by continuous functions.

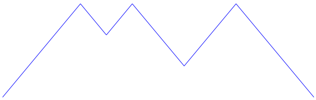
Given any continuous excursion, we can define an associated tree (metric space), by gluing together points of the excursion that (a) have the same height and (b) can see each other.



[Picture by Christina Goldschmidt.]

Trees coded by continuous functions.

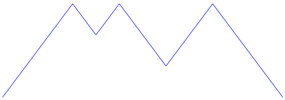
Given any continuous excursion, we can define an associated tree (metric space), by gluing together points of the excursion that (a) have the same height and (b) can see each other.



[Picture by Christina Goldschmidt.]

Trees coded by continuous functions.

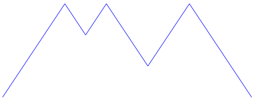
Given any continuous excursion, we can define an associated tree (metric space), by gluing together points of the excursion that (a) have the same height and (b) can see each other.



[Picture by Christina Goldschmidt.]

Trees coded by continuous functions.

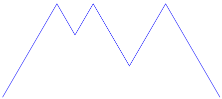
Given any continuous excursion, we can define an associated tree (metric space), by gluing together points of the excursion that (a) have the same height and (b) can see each other.



[Picture by Christina Goldschmidt.]

Trees coded by continuous functions.

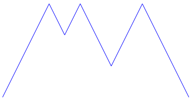
Given any continuous excursion, we can define an associated tree (metric space), by gluing together points of the excursion that (a) have the same height and (b) can see each other.



[Picture by Christina Goldschmidt.]

Trees coded by continuous functions.

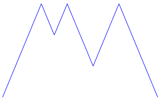
Given any continuous excursion, we can define an associated tree (metric space), by gluing together points of the excursion that (a) have the same height and (b) can see each other.



[Picture by Christina Goldschmidt.]

Trees coded by continuous functions.

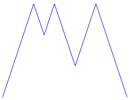
Given any continuous excursion, we can define an associated tree (metric space), by gluing together points of the excursion that (a) have the same height and (b) can see each other.



[Picture by Christina Goldschmidt.]

Trees coded by continuous functions.

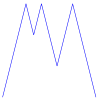
Given any continuous excursion, we can define an associated tree (metric space), by gluing together points of the excursion that (a) have the same height and (b) can see each other.



[Picture by Christina Goldschmidt.]

Trees coded by continuous functions.

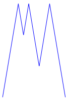
Given any continuous excursion, we can define an associated tree (metric space), by gluing together points of the excursion that (a) have the same height and (b) can see each other.



[Picture by Christina Goldschmidt.]

Trees coded by continuous functions.

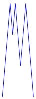
Given any continuous excursion, we can define an associated tree (metric space), by gluing together points of the excursion that (a) have the same height and (b) can see each other.



[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.]

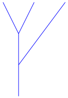
Trees coded by continuous functions.

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.

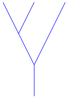
Given any continuous excursion, we can define an associated tree (metric space), by gluing together points of the excursion that (a) have the same height and (b) can see each other.



[Picture by Christina Goldschmidt.]

Trees coded by continuous functions.

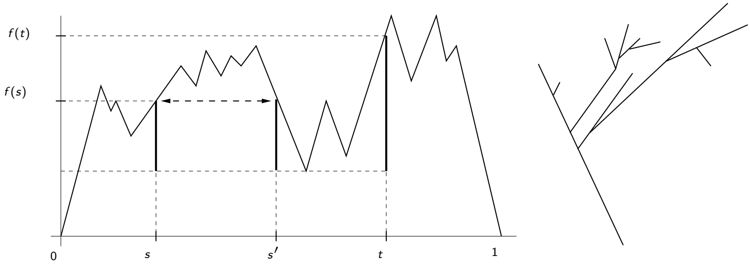
Given any continuous excursion, we can define an associated tree (metric space), by gluing together points of the excursion that (a) have the same height and (b) can see each other.



[Picture by Christina Goldschmidt.]

Trees coded by continuous functions.

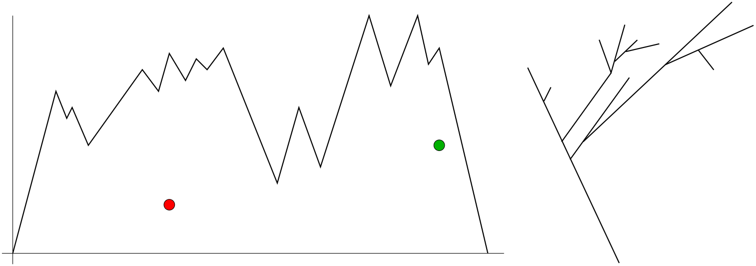
Given any continuous excursion, we can define an associated tree (metric space), by gluing together points of the excursion that (a) have the same height and (b) can see each other.



[Picture by Marie Albenque.]

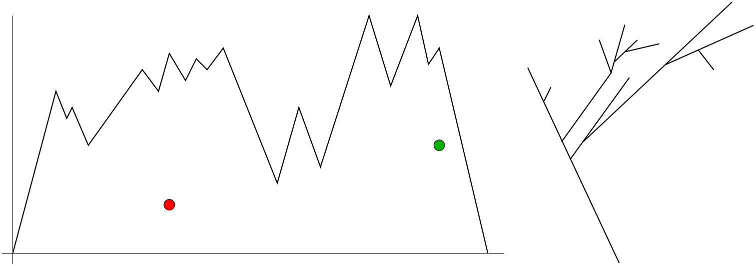
Handling the surplus edges.

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.

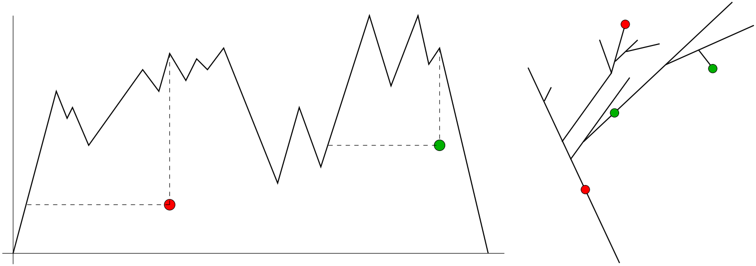
We saw how to associate a tree to a continuous excursion by gluing together points of the excursion that (a) have the same height and (b) can see each other.



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

Handling the surplus edges.

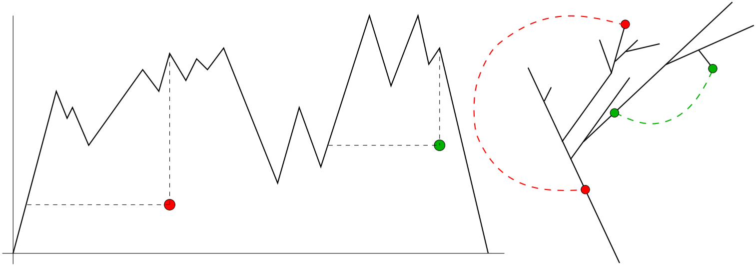
We saw how to associate a tree to a continuous excursion by gluing together points of the excursion that (a) have the same height and (b) can see each other.



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 $1, 2, \ldots, m$.

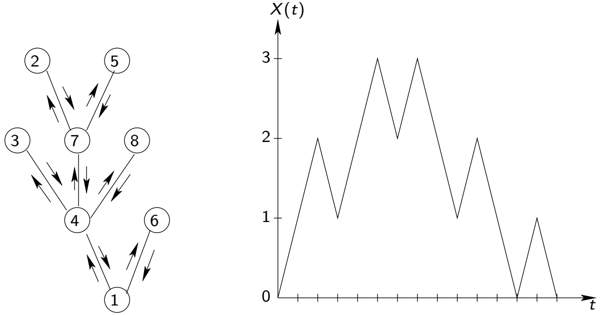
Uniform random trees.

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.

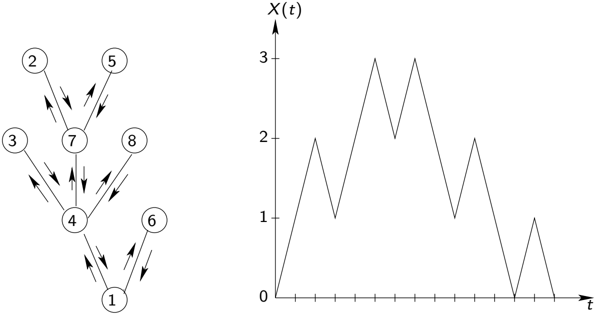
The Harris walk is an excursion encoding heights in a finite tree.



[Picture by Marie Albenque.]

The Harris walk.

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



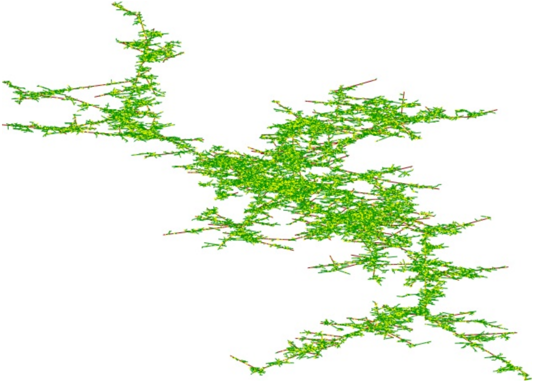
[Picture by Marie Albenque.]

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.

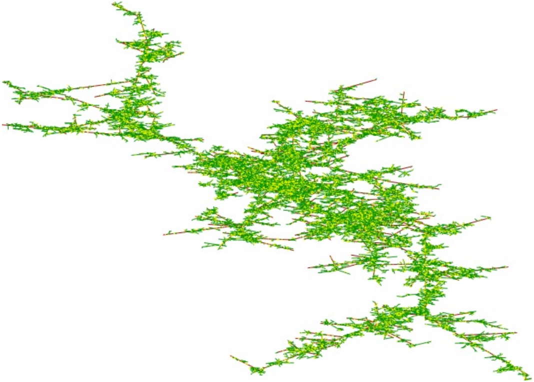
If the *tree itself* is rescaled (edge lengths $1/\sqrt{m}$ instead of 1), then as $m \to \infty$, there is a limiting object, called the Brownian continuum random tree.



[Picture by Grégory Miermont.]

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.



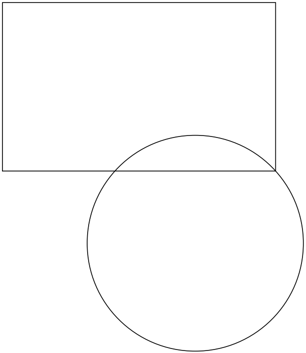
[Picture by Grégory Miermont.]

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

Metric space convergence.

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

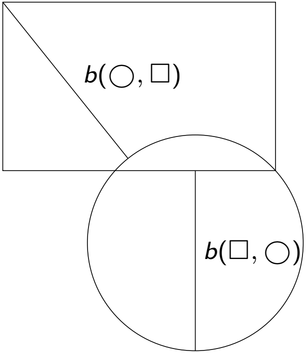
$$b(S, T) = \inf\{r \;:\; T \subset b_r(S)\}.$$



Metric space convergence.

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

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



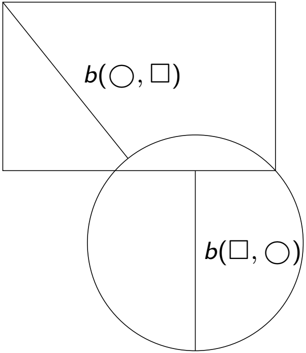
Metric space convergence.

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 $\max(b(S, T), b(T, S))$.



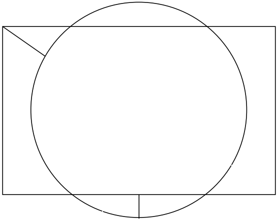
Metric space convergence.

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 $\max(b(S, T), b(T, S))$.



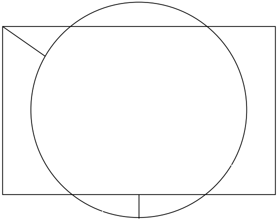
Metric space convergence.

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.

Metric space convergence.

The uniform random tree on $1, \ldots, m$, with distances rescaled by $1/\sqrt{m}$, converges in distribution to the Brownian continuum random tree, with respect to the Gromov-Hausdorff distance.

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 $G_{n,1/n}$.

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 $G_{n,1/n}$.
We prove 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 $G_{n,1/n}$.
We prove that

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

The sequence of metric spaces $(\mathcal{M}_1, \mathcal{M}_2, \ldots)$ is obtained from Aldous's reflected Brownian motion with drift by making the identifications given by the Poisson process.

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 $G_{n,1/n}$.
We prove that

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

The sequence of metric spaces $(\mathcal{M}_1, \mathcal{M}_2, \ldots)$ is obtained from Aldous's reflected Brownian motion with drift by making the identifications given by the Poisson process.

Here convergence is with respect to the metric

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

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

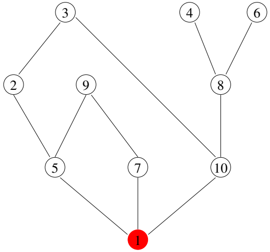
The canonical excursion is constructed via a procedure called depth-first exploration.

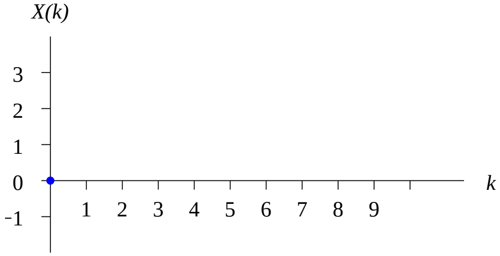
We will explore the graph step-by-step. At each step, vertices will have the possibility to be current, alive or dead.

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



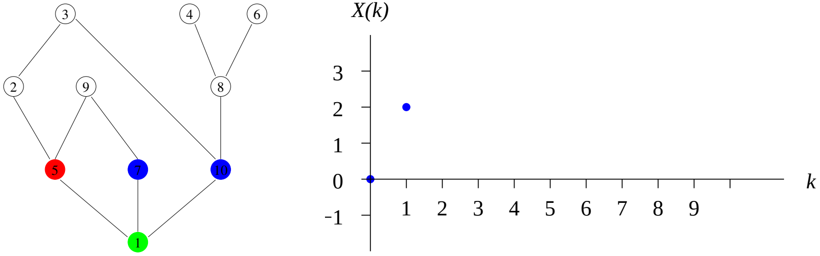


[Picture by Christina Goldschmidt.]

Current: 1 Alive: none Dead: none

Depth-first walk

Step 1

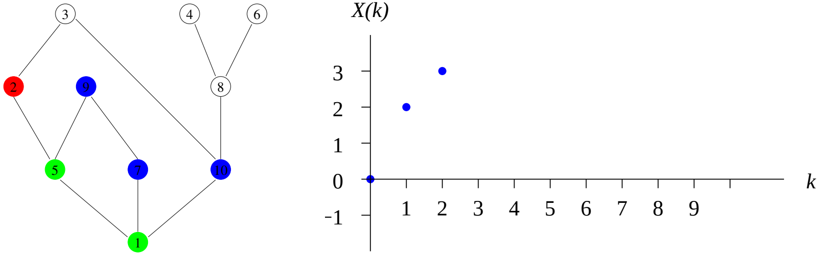


[Picture by Christina Goldschmidt.]

Current: 5 Alive: 7,10 Dead: 1

Depth-first walk

Step 2

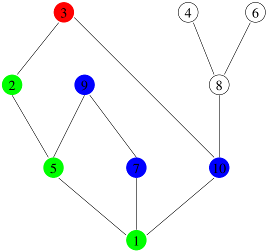


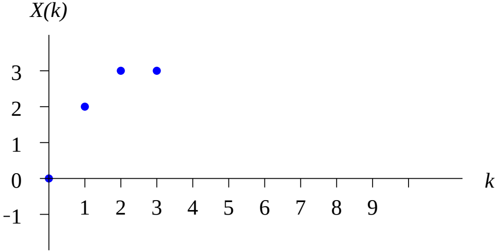
[Picture by Christina Goldschmidt.]

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

Depth-first walk

Step 3



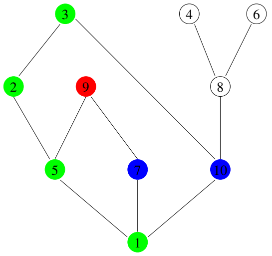


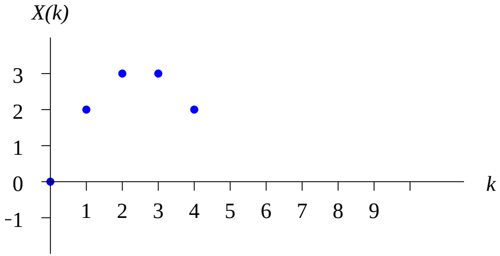
[Picture by Christina Goldschmidt.]

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

Depth-first walk

Step 4



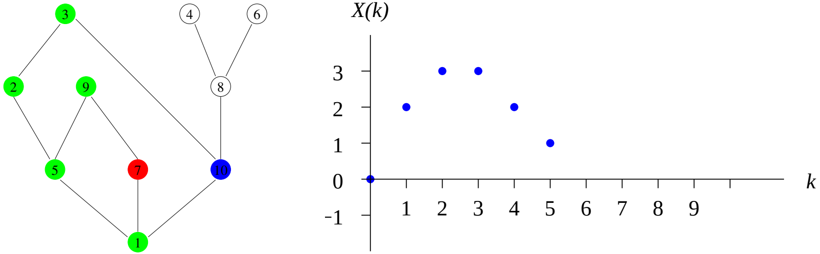


[Picture by Christina Goldschmidt.]

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

Depth-first walk

Step 5

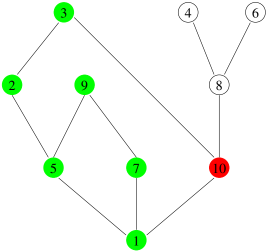


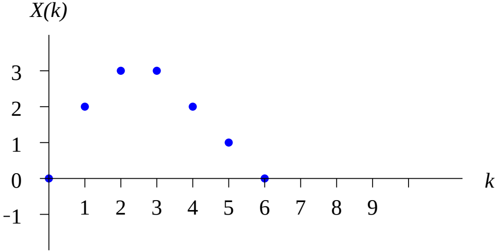
[Picture by Christina Goldschmidt.]

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

Depth-first walk

Step 6



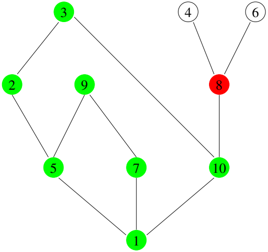


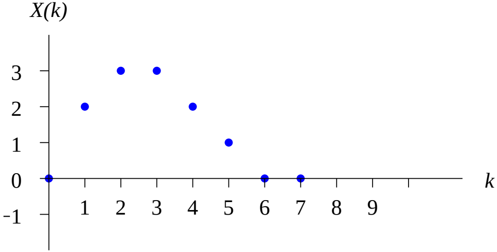
[Picture by Christina Goldschmidt.]

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

Depth-first walk

Step 7



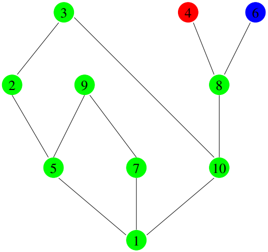


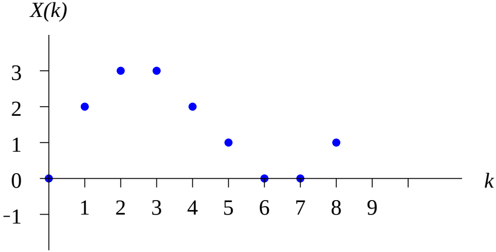
[Picture by Christina Goldschmidt.]

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

Depth-first walk

Step 8



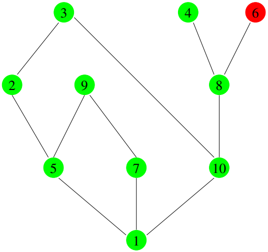


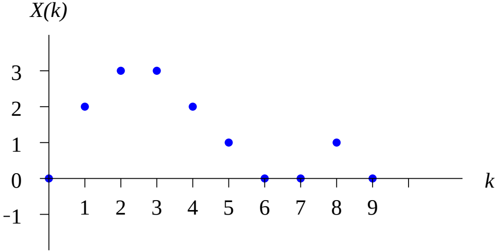
[Picture by Christina Goldschmidt.]

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

Depth-first walk

Step 9

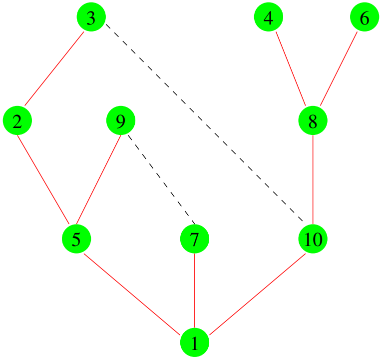




[Picture by Christina Goldschmidt.]

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?

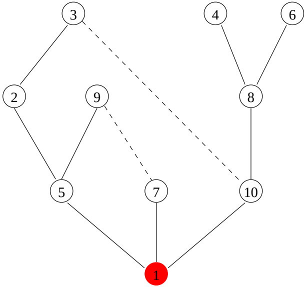
Permitted edges

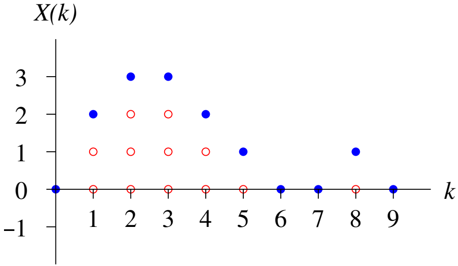
What are the set of graphs with a given depth-first tree/canonical discrete excursion?

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

Call such edges permitted.

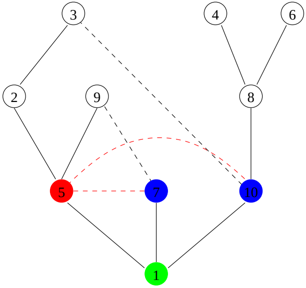
Depth-first walk and permitted edges

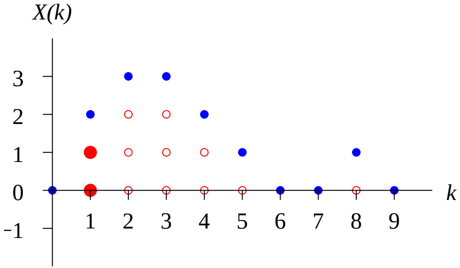




Step 0: $X(0) = 0$.

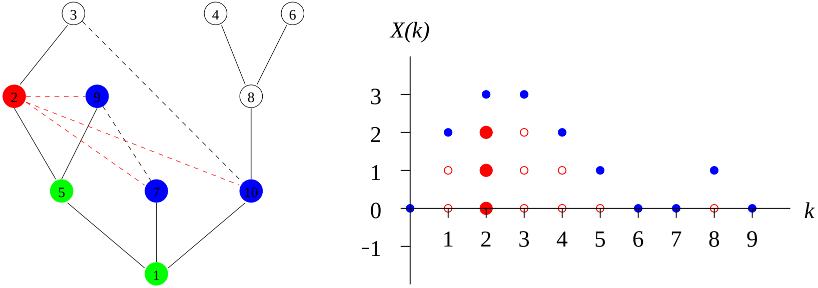
Depth-first walk and permitted edges





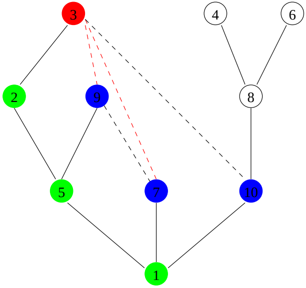
Step 1: $X(1) = 2$.

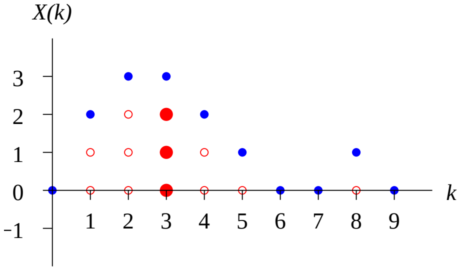
Depth-first walk and permitted edges



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

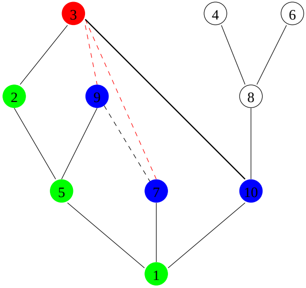
Depth-first walk and permitted edges

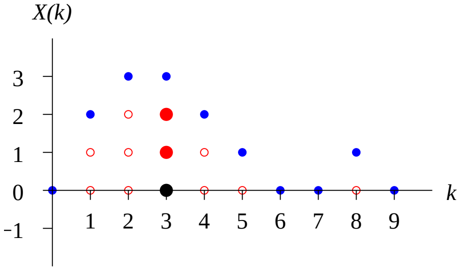




Step 3: $X(3) = 3$.

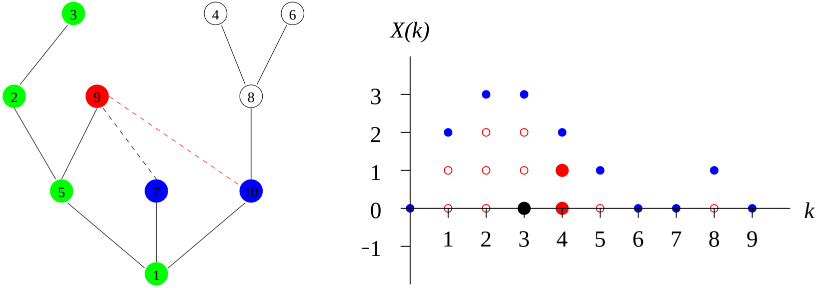
Depth-first walk and permitted edges





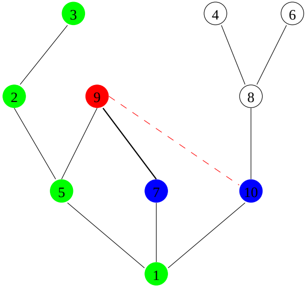
Step 3: $X(3) = 3$.

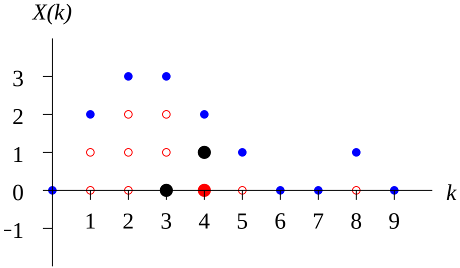
Depth-first walk and permitted edges



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

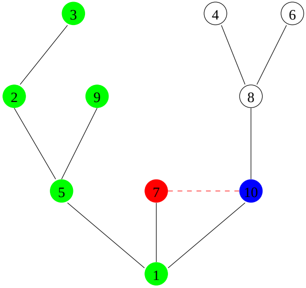
Depth-first walk and permitted edges

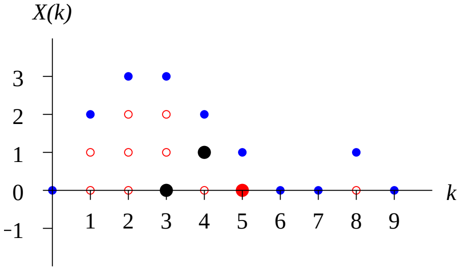




Step 4: $X(4) = 2$.

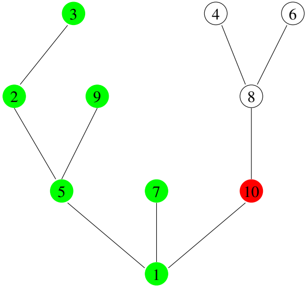
Depth-first walk and permitted edges

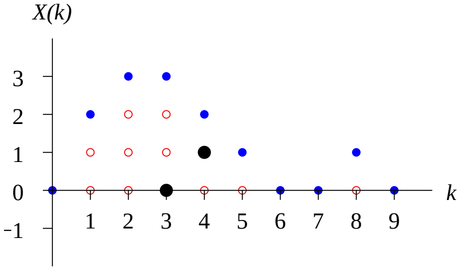




Step 5: $X(5) = 1$.

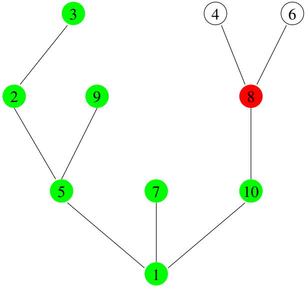
Depth-first walk and permitted edges

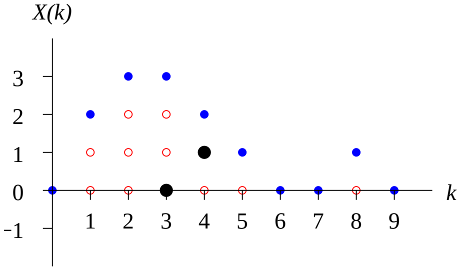




Step 6: $X(6) = 0$.

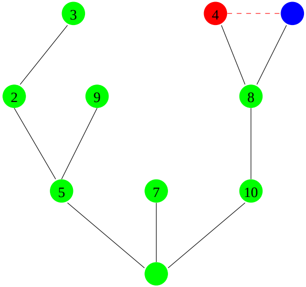
Depth-first walk and permitted edges

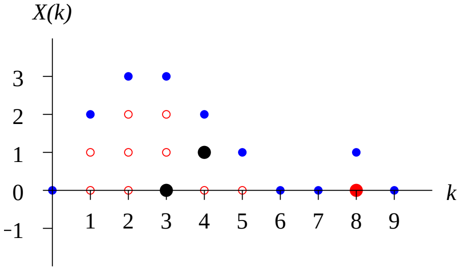




Step 7: $X(7) = 0$.

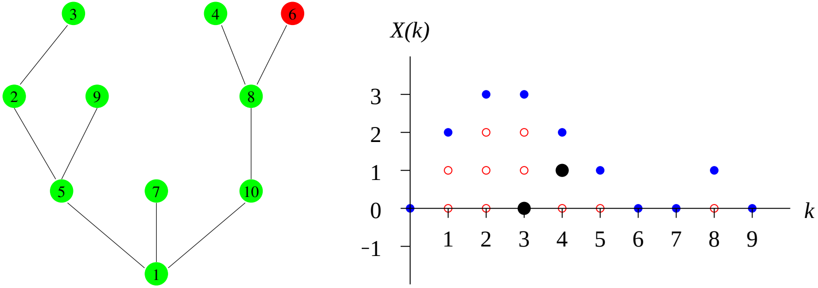
Depth-first walk and permitted edges





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

Depth-first walk and permitted edges



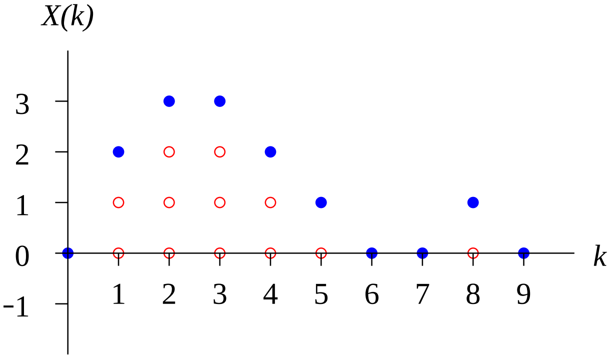
Step 10: $X(9) = 0$.

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 $[m] = \{1, 2, \dots, m\}$. 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 $1, \dots, m$. 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 $1, 2, \dots, m$ 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.

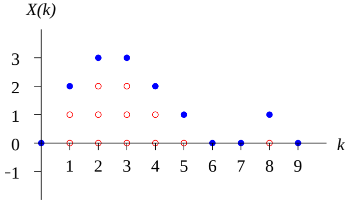
We wanted a way of generating a random component of $G_{n,1/n}$ with size m. We are now in a position to do this.

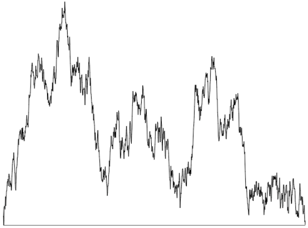
- First select a tree randomly by picking each of the labelled trees T on $1, 2, \dots, m$ with a probability weight proportional to $(1 - 1/n)^{-a(T)}$.
- Having chosen T, then add each of the $a(T)$ allowed surplus edges (marks) independently with probability $1/n$.

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

The scaling limit of depth-first walks.

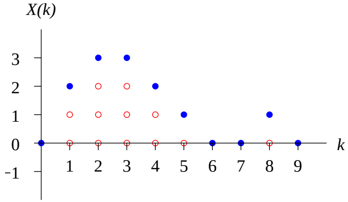
When time is scaled by $1/m$, space is scaled by $1/\sqrt{m}$, the limit of depth-first walk of a *uniform* tree is *Brownian excursion*.

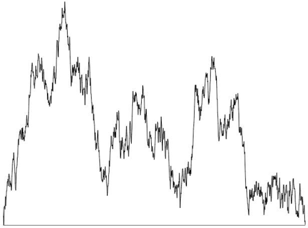




The scaling limit of depth-first walks.

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*.

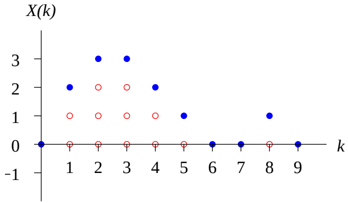


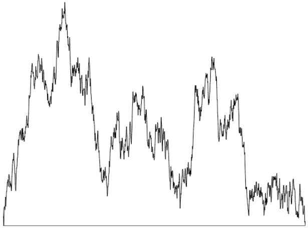


The *area* $a(T)$ is then $\Theta(m^{3/2})$, so if $m = n^{2/3}$ then $a(T) = \Theta(n)$ and

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

The scaling limit of depth-first walks.

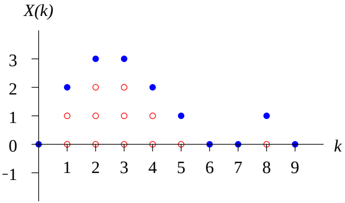


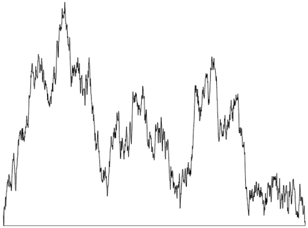


$$\left(1 - \frac{1}{n}\right)^{-a(T)} \sim e^{-a(T)/n}.$$

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

The scaling limit of depth-first walks.



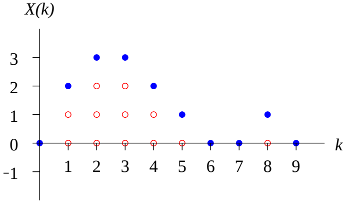


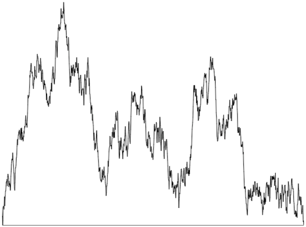
$$\left(1-\frac{1}{n}\right)^{-a(T)} \sim e^{-a(T)/n}.$$

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.





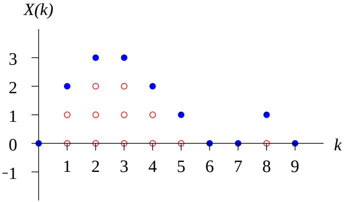
$$\left(1-\frac{1}{n}\right)^{-a(T)} \sim e^{-a(T)/n}.$$

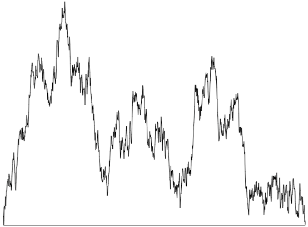
We choose a tree with probability prop. to $(1-1/n)^{-a(T)}$.

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.

The last slide, I promise.

The exponential tilt of a Brownian excursion is well-defined since the maximum of a Brownian excursion has sub-Gaussian tails.

The last slide, I promise.

The exponential tilt of a Brownian excursion is well-defined since the maximum of a Brownian excursion has sub-Gaussian tails.

For proving the above convergence, we needed similar tail bounds for the height h_n of a uniformly random tree with n nodes. Luckily, sub-Gaussian tails for $h_n/\sqrt{n}$ were already known (Flajolet & Odlyzko, 1982).

The last slide, I promise.

The exponential tilt of a Brownian excursion is well-defined since the maximum of a Brownian excursion has sub-Gaussian tails.

For proving the above convergence, we needed similar tail bounds for the height h_n of a uniformly random tree with n nodes. Luckily, sub-Gaussian tails for $h_n/\sqrt{n}$ were already known (Flajolet & Odlyzko, 1982).

A uniformly random tree with n nodes has the same distribution as a Poisson Galton-Watson tree conditioned to have n nodes.

The last slide, I promise.

The exponential tilt of a Brownian excursion is well-defined since the maximum of a Brownian excursion has sub-Gaussian tails.

For proving the above convergence, we needed similar tail bounds for the height h_n of a uniformly random tree with n nodes. Luckily, sub-Gaussian tails for $h_n/\sqrt{n}$ were already known (Flajolet & Odlyzko, 1982).

A uniformly random tree with n nodes has the same distribution as a Poisson Galton-Watson tree conditioned to have n nodes.

Fix any branching distribution B with $\mathbb{P}(B > 1) > 0$. Let T be a Galton-Watson tree branching distribution B, conditioned to have n nodes, and let $h(T)$ be its height.

The last slide, I promise.

The exponential tilt of a Brownian excursion is well-defined since the maximum of a Brownian excursion has sub-Gaussian tails.

For proving the above convergence, we needed similar tail bounds for the height h_n of a uniformly random tree with n nodes. Luckily, sub-Gaussian tails for $h_n/\sqrt{n}$ were already known (Flajolet & Odlyzko, 1982).

A uniformly random tree with n nodes has the same distribution as a Poisson Galton-Watson tree conditioned to have n nodes.

Fix any branching distribution B with $\mathbb{P}(B > 1) > 0$. Let T be a Galton-Watson tree branching distribution B, conditioned to have n nodes, and let $h(T)$ be its height.

Question (A version of a question asked by Svante Janson)

Is it true that $h(T)/\sqrt{n}$ has sub-Gaussian tails, uniformly in n?