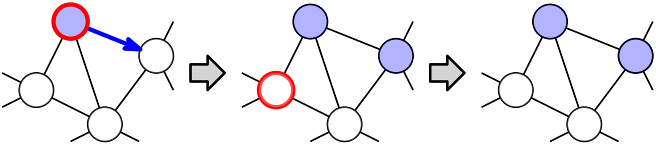
Evolutionary graph theory

Josef "Pepa" Tkadlec

Mathematical Forum, March 2026



Warm-up puzzle

Puzzle. How many times do you need to roll a die, on average, until number 5 appears?

Warm-up puzzle

Puzzle. How many times do you need to roll a die, on average, until number 5 appears?

Answer:

Warm-up puzzle

Puzzle. How many times do you need to roll a die, on average, until number 5 appears?

Answer: 6 times.

Solution.

Warm-up puzzle

Puzzle. How many times do you need to roll a die, on average, until number 5 appears?

Answer: 6 times.

Solution. Denote the answer by x. Then

$$x = 1 + \frac{1}{6} \cdot 0 + \frac{5}{6} \cdot x.$$

This gives $x = 6$.

Warm-up puzzle

Puzzle. How many times do you need to roll a die, on average, until number 5 appears?

Answer: 6 times.

Solution. Denote the answer by x. Then

$$x = 1 + \frac{1}{6} \cdot 0 + \frac{5}{6} \cdot x.$$

This gives $x = 6$.

Jug lemma: "On average, an event with probability p happens after $1/p$ steps."

How does stuff propagate through networks?

COVID pandemic

- Nodes: people. Edges: spending time together
- Process: people getting Susceptible → Infected → Recovered

Questions

- Which region gets hit next? When?
- Is there enough hospital capacity?
- Who should we vaccinate first?
- ...

How does stuff propagate through networks?

- epidemiology: coronavirus among humans
- social dynamics: influence (opinion, gossip, fake news) on social media
- evolutionary biology: genetic mutations in populations
- statistical mechanics, percolation, distributed computing, ...

System = a bunch of nodes of different types.

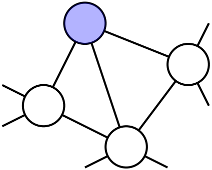
Overview

- PUSH protocol for rumor spreading
- Moran process on graphs

PUSH protocol for rumor spreading

[Eden '61] [Boyd '06] Given a graph $G_n = (V, E)$ on n nodes.

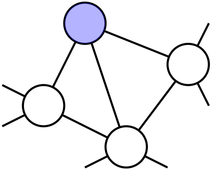
- Initially: one node v is informed (blue) rest is not (white)



PUSH protocol for rumor spreading

[Eden '61] [Boyd '06] Given a graph $G_n = (V, E)$ on n nodes.

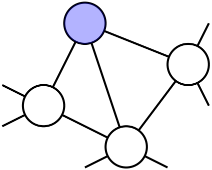
- Initially: one node v is informed (blue) rest is not (white)
- In each round, repeat:



PUSH protocol for rumor spreading

[Eden '61] [Boyd '06] Given a graph $G_n = (V, E)$ on n nodes.

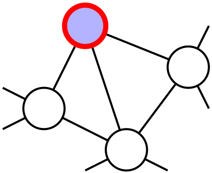
- Initially: one node v is informed (blue) rest is not (white)
- In each round, repeat:
  1. Pick a random node



PUSH protocol for rumor spreading

[Eden '61] [Boyd '06] Given a graph $G_n = (V, E)$ on n nodes.

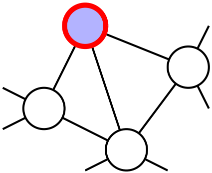
- Initially: one node v is informed (blue) rest is not (white)
- In each round, repeat:
  1. Pick a random node



PUSH protocol for rumor spreading

[Eden '61] [Boyd '06] Given a graph $G_n = (V, E)$ on n nodes.

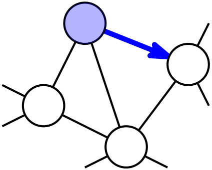
- Initially: one node v is informed (blue) rest is not (white)
- In each round, repeat:
  1. Pick a random node
  2. If it is informed, make a random neighbor informed



PUSH protocol for rumor spreading

[Eden '61] [Boyd '06] Given a graph $G_n = (V, E)$ on n nodes.

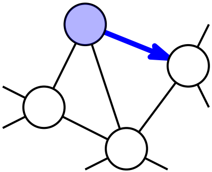
- Initially: one node v is informed (blue) rest is not (white)
- In each round, repeat:
  1. Pick a random node
  2. If it is informed, make a random neighbor informed



PUSH protocol for rumor spreading

[Eden '61] [Boyd '06] Given a graph $G_n = (V, E)$ on n nodes.

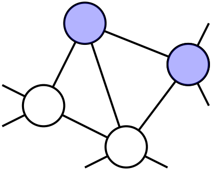
- Initially: one node v is informed (blue) rest is not (white)
- In each round, repeat:
  1. Pick a random node
  2. If it is informed, make a random neighbor informed



PUSH protocol for rumor spreading

[Eden '61] [Boyd '06] Given a graph $G_n = (V, E)$ on n nodes.

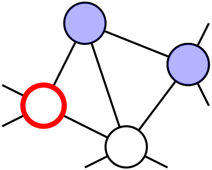
- Initially: one node v is informed (blue) rest is not (white)
- In each round, repeat:
  1. Pick a random node
  2. If it is informed, make a random neighbor informed



PUSH protocol for rumor spreading

[Eden '61] [Boyd '06] Given a graph $G_n = (V, E)$ on n nodes.

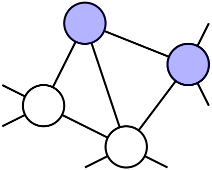
- Initially: one node v is informed (blue) rest is not (white)
- In each round, repeat:
    1. Pick a random node
    2. If it is informed, make a random neighbor informed



PUSH protocol for rumor spreading

[Eden '61] [Boyd '06] Given a graph $G_n = (V, E)$ on n nodes.

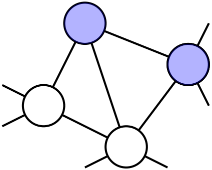
- Initially: one node v is informed (blue) rest is not (white)
- In each round, repeat:
  1. Pick a random node
  2. If it is informed, make a random neighbor informed



PUSH protocol for rumor spreading

[Eden '61] [Boyd '06] Given a graph $G_n = (V, E)$ on n nodes.

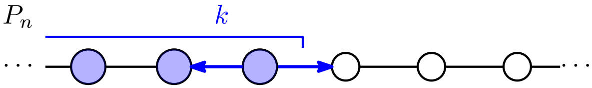
- Initially: one node v is informed (blue) rest is not (white)
- In each round, repeat:
  1. Pick a random node
  2. If it is informed, make a random neighbor informed



Question: How long until all nodes are informed?

Denote the expected number of rounds by $T(G_n, v)$.

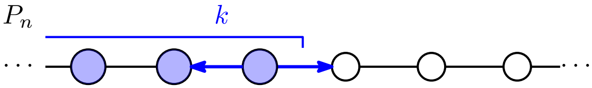
Path graph P_n



- Idea: Wait for $n-1$ stages.
- Let $p_k =$"chance to make progress, with k nodes informed".
- On average, progress after $T_k = \frac{1}{p_k}$ rounds [Puzzle].
- In total:

$$T(P_n, v) = \cdots + \frac{1}{\frac{1}{n} \cdot \frac{1}{2}} + \ldots$$

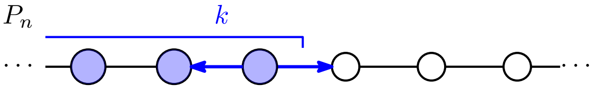
Path graph P_n



- Idea: Wait for $n-1$ stages.
- Let $p_k =$"chance to make progress, with k nodes informed".
- On average, progress after $T_k = \frac{1}{p_k}$ rounds [Puzzle].
- In total:

$$T(P_n, v) = \dots + \frac{1}{\frac{1}{n} \cdot \frac{1}{2}} + \dots$$

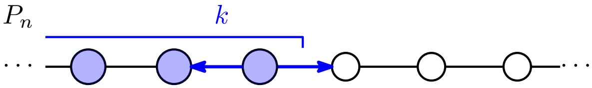
Path graph P_n



- Idea: Wait for $n-1$ stages.
- Let $p_k =$"chance to make progress, with k nodes informed".
- On average, progress after $T_k = \frac{1}{p_k}$ rounds [Puzzle].
- In total:

$$T(P_n, v) = \dots + \frac{1}{\frac{1}{n} \cdot \frac{1}{2}} + \dots = \dots + 2n + \dots =$$
$$= 2n^2 - 3n = 2n^2 + o(n^2) \approx 2n^2 = \Theta(n^2).$$

Path graph P_n

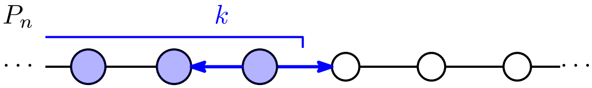


- Idea: Wait for $n-1$ stages.
- Let $p_k =$ "chance to make progress, with k nodes informed".
- On average, progress after $T_k = \frac{1}{p_k}$ rounds [Puzzle].
- In total:

$$T(P_n, v) = \dots + \frac{1}{\frac{1}{n} \cdot \frac{1}{2}} + \dots = \dots + 2n + \dots =$$
$$= 2n^2 - 3n = 2n^2 + o(n^2) \approx 2n^2 = \Theta(n^2).$$

On a Cycle graph C_n: $T(C_n, v) = \dots$

Path graph P_n

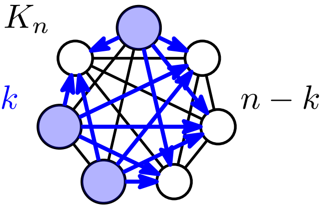


- Idea: Wait for $n-1$ stages.
- Let $p_k =$"chance to make progress, with k nodes informed".
- On average, progress after $T_k = \frac{1}{p_k}$ rounds [Puzzle].
- In total:

$$T(P_n, v) = \dots + \frac{1}{\frac{1}{n}\cdot\frac{1}{2}} + \dots = \dots + 2n + \dots =$$
$$= 2n^2 - 3n = 2n^2 + o(n^2) \approx 2n^2 = \Theta(n^2).$$

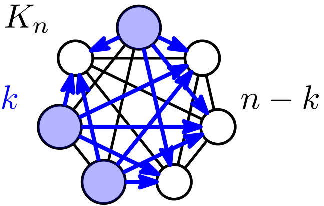
On a Cycle graph C_n: $T(C_n, v) = \dots = n(n-1) \approx n^2$.

Complete graph K_n



- Same idea: $p_k = \dots$

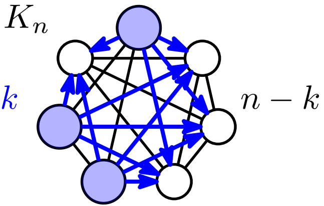
Complete graph K_n



- Same idea: $p_k = \ldots \frac{k}{n} \cdot \frac{n-k}{n-1}$
- Total time

$$T(K_n, v) =$$

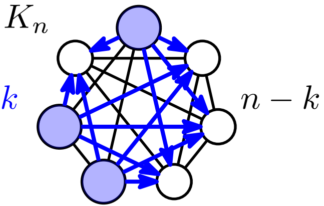
Complete graph K_n



- Same idea: $p_k = \dots \frac{k}{n} \cdot \frac{n-k}{n-1}$
- Total time

$$T(K_n, v) = \sum_{k=1}^{n-1} \frac{n(n-1)}{k(n-k)} =$$

Complete graph K_n

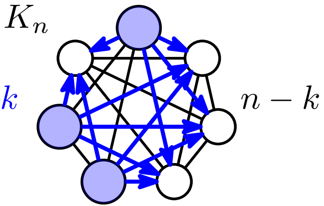


- Same idea: $p_k = \dots \frac{k}{n} \cdot \frac{n-k}{n-1}$
- Total time

$$T(K_n, v) = \sum_{k=1}^{n-1} \frac{n(n-1)}{k(n-k)} = (n-1) \cdot \sum_{k=1}^{n-1} \frac{k+n-k}{k(n-k)}$$

$$=$$

Complete graph K_n

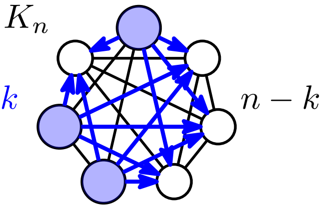


- Same idea: $p_k = \ldots \frac{k}{n} \cdot \frac{n-k}{n-1}$
- Total time

$$T(K_n, v) = \sum_{k=1}^{n-1} \frac{n(n-1)}{k(n-k)} = (n-1) \cdot \sum_{k=1}^{n-1} \frac{k+n-k}{k(n-k)}$$

$$= 2(n-1) \cdot \left(\frac{1}{1} + \frac{1}{2} + \cdots + \frac{1}{n-1} \right) \approx$$

Complete graph K_n

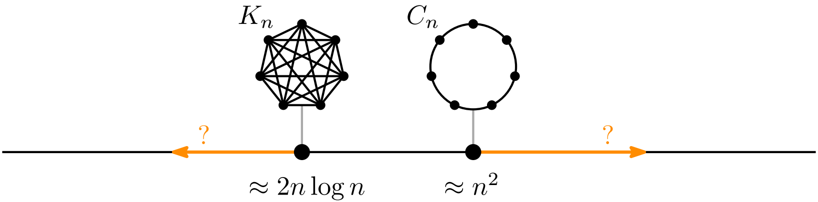


- Same idea: $p_k = \dots \frac{k}{n} \cdot \frac{n-k}{n-1}$
- Total time

$$T(K_n, v) = \sum_{k=1}^{n-1} \frac{n(n-1)}{k(n-k)} = (n-1) \cdot \sum_{k=1}^{n-1} \frac{k+n-k}{k(n-k)}$$

$$= 2(n-1) \cdot \left(\frac{1}{1} + \frac{1}{2} + \cdots + \frac{1}{n-1} \right) \approx 2n \log n.$$

How fast / slow can a graph be?



Puzzle: Coupon collector problem

Puzzle. On average, how many times do you need to roll a die until each of the numbers 1 to 6 appears at least once?

Puzzle: Coupon collector problem

Puzzle. On average, how many times do you need to roll a die until each of the numbers 1 to 6 appears at least once?

Answer:

Puzzle: Coupon collector problem

Puzzle. On average, how many times do you need to roll a die until each of the numbers 1 to 6 appears at least once?

Answer: 14.7 times.

Puzzle: Coupon collector problem

Puzzle. On average, how many times do you need to roll a die until each of the numbers 1 to 6 appears at least once?

Answer: 14.7 times.

Solution. The first roll gives a number.
The next roll gives a “new” number with probability $\frac{5}{6}$.
$\rightarrow$ on average, we get the second number after $\frac{6}{5}$ rolls.
Likewise for the third number and so on. The answer is:

Puzzle: Coupon collector problem

Puzzle. On average, how many times do you need to roll a die until each of the numbers 1 to 6 appears at least once?

Answer: 14.7 times.

Solution. The first roll gives a number.
The next roll gives a "new" number with probability $\frac{5}{6}$.
$\rightarrow$ on average, we get the second number after $\frac{6}{5}$ rolls.
Likewise for the third number and so on. The answer is:

$$1 + \frac{1}{5/6} + \frac{1}{4/6} + \cdots + \frac{1}{1/6} = \frac{6}{6} + \frac{6}{5} + \cdots + \frac{6}{1} = 14.7.$$

Puzzle: Coupon collector problem

Puzzle. On average, how many times do you need to roll a die until each of the numbers 1 to 6 appears at least once?

Answer: 14.7 times.

Solution. The first roll gives a number.
The next roll gives a "new" number with probability $\frac{5}{6}$.
$\rightarrow$ on average, we get the second number after $\frac{6}{5}$ rolls.
Likewise for the third number and so on. The answer is:

$$1 + \frac{1}{5/6} + \frac{1}{4/6} + \cdots + \frac{1}{1/6} = \frac{6}{6} + \frac{6}{5} + \cdots + \frac{6}{1} = 14.7.$$

For a fair n-sided die, the answer is $\frac{n}{n} + \frac{n}{n-1} + \cdots + \frac{n}{1} \approx$

Puzzle: Coupon collector problem

Puzzle. On average, how many times do you need to roll a die until each of the numbers 1 to 6 appears at least once?

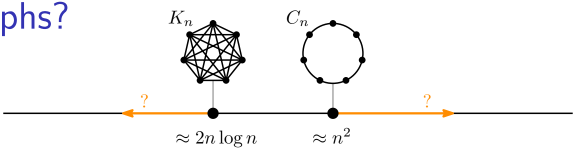
Answer: 14.7 times.

Solution. The first roll gives a number.
The next roll gives a "new" number with probability $\frac{5}{6}$.
$\rightarrow$ on average, we get the second number after $\frac{6}{5}$ rolls.
Likewise for the third number and so on. The answer is:

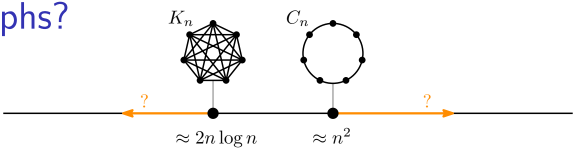
$$1 + \frac{1}{5/6} + \frac{1}{4/6} + \cdots + \frac{1}{1/6} = \frac{6}{6} + \frac{6}{5} + \cdots + \frac{6}{1} = 14.7.$$

For a fair n-sided die, the answer is $\frac{n}{n} + \frac{n}{n-1} + \cdots + \frac{n}{1} \approx n \log n$.

Fastest graphs?

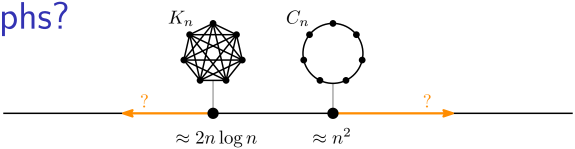


Fastest graphs?



Claim. For any G_n, v we have $T(G_n, v) \geq n \log n + o(n \log n)$.

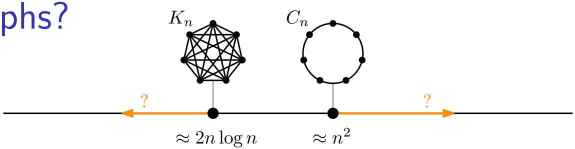
Fastest graphs?



Claim. For any G_n, v we have $T(G_n, v) \geq n \log n + o(n \log n)$.

Proof idea. Each node except v must become informed. But that's like rolling an (unfair) die and waiting for $n-1$ different numbers to appear.

Fastest graphs?

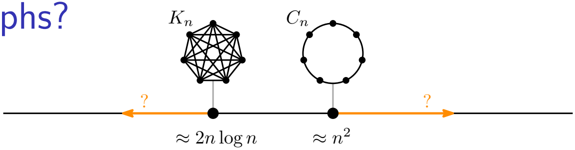


Claim. For any G_n, v we have $T(G_n, v) \geq n \log n + o(n \log n)$.

Proof idea. Each node except v must become informed. But that's like rolling an (unfair) die and waiting for $n-1$ different numbers to appear.

Claim. In fact, $T(G_n, v) \geq 2n \log n + o(n \log n)$.

Fastest graphs?



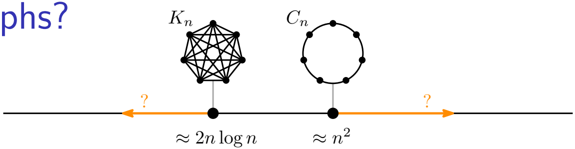
Claim. For any G_n, v we have $T(G_n, v) \geq n \log n + o(n \log n)$.

Proof idea. Each node except v must become informed. But that's like rolling an (unfair) die and waiting for $n-1$ different numbers to appear.

Claim. In fact, $T(G_n, v) \geq 2n \log n + o(n \log n)$.

Conjecture ['20]. Complete graph K_n is the fastest graph.

Fastest graphs?



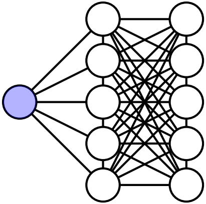
Claim. For any G_n, v we have $T(G_n, v) \geq n \log n + o(n \log n)$.

Proof idea. Each node except v must become informed. But that's like rolling an (unfair) die and waiting for $n - 1$ different numbers to appear.

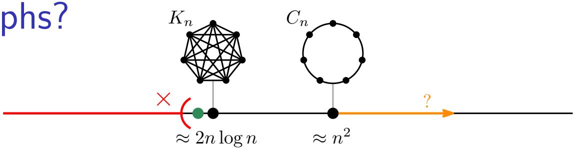
Claim. In fact, $T(G_n, v) \geq 2n \log n + o(n \log n)$.

Conjecture ['20]. Complete graph K_n is the fastest graph.

Claim [KT, '24]. Actually no – there are (slightly) faster graphs and starting nodes!



Fastest graphs?



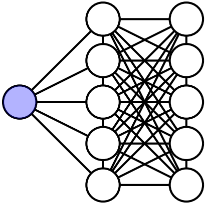
Claim. For any G_n, v we have $T(G_n, v) \geq n \log n + o(n \log n)$.

Proof idea. Each node except v must become informed. But that's like rolling an (unfair) die and waiting for $n - 1$ different numbers to appear.

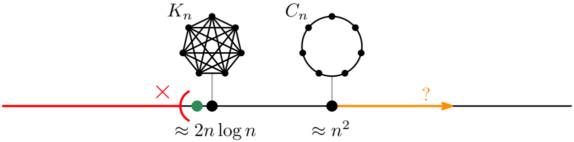
Claim. In fact, $T(G_n, v) \geq 2n \log n + o(n \log n)$.

Conjecture ['20]. Complete graph K_n is the fastest graph.

Claim [KT, '24]. Actually no – there are (slightly) faster graphs and starting nodes!

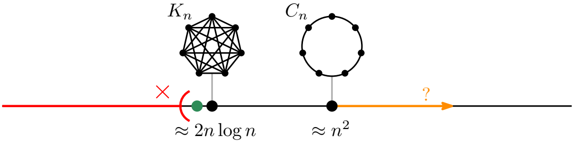


Slowest graphs?



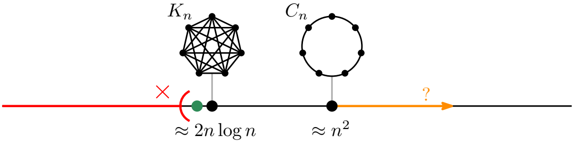
Claim (Upper bound). We have $T(G_n, v) \leq \ldots$

Slowest graphs?



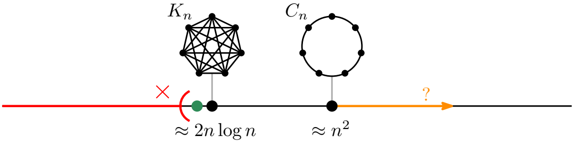
Claim (Upper bound). We have $T(G_n, v) \leq \ldots n^3$.

Slowest graphs?



Claim (Upper bound). We have $T(G_n, v) \leq \ldots n^3$.
Proof idea: Because $T_k \leq n^2$.

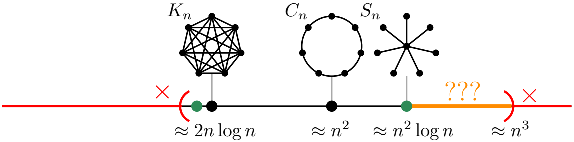
Slowest graphs?



Claim (Upper bound). We have $T(G_n, v) \leq \ldots n^3$.
Proof idea: Because $T_k \leq n^2$.

Claim. For the Star S_n we have $T(S_n, v) \approx n^2 \log n$.

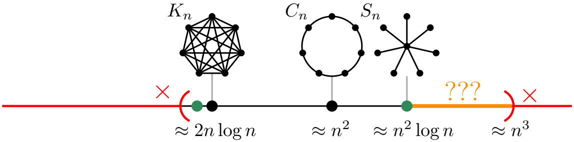
Slowest graphs?



Claim (Upper bound). We have $T(G_n, v) \leq \ldots n^3$.
Proof idea: Because $T_k \leq n^2$.

Claim. For the Star S_n we have $T(S_n, v) \approx n^2 \log n$.

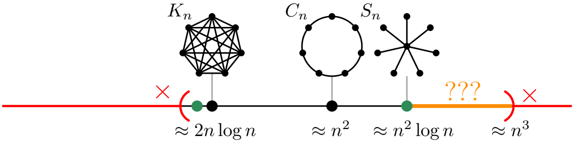
Slowest graphs?



Claim (Upper bound). We have $T(G_n, v) \leq \ldots n^3$.
Proof idea: Because $T_k \leq n^2$.

Claim. For the Star S_n we have $T(S_n, v) \approx n^2 \log n$.
Proof idea: Like collecting coupons, but only once every n steps.

Slowest graphs?



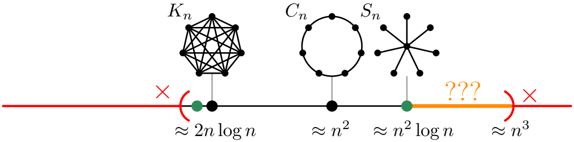
Claim (Upper bound). We have $T(G_n, v) \leq \ldots n^3$.
Proof idea: Because $T_k \leq n^2$.

Claim. For the Star S_n we have $T(S_n, v) \approx n^2 \log n$.
Proof idea: Like collecting coupons, but only once every n steps.

Claim [Feige '90]. In fact, $T(G_n, v) = \mathcal{O}(n^2 \log n)$.

Question. Is S_n the slowest graph?

Slowest graphs?



Claim (Upper bound). We have $T(G_n, v) \leq \ldots n^3$.
Proof idea: Because $T_k \leq n^2$.

Claim. For the Star S_n we have $T(S_n, v) \approx n^2 \log n$.
Proof idea: Like collecting coupons, but only once every n steps.

Claim [Feige '90]. In fact, $T(G_n, v) = \mathcal{O}(n^2 \log n)$.

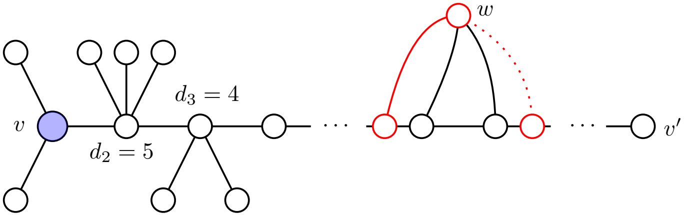
Question. Is S_n the slowest graph? This is Open.

Stronger upper bound

Key lemma. For any G_n, v, v' we have $T(v \to v') \leq 5n^2$.

Proof idea:

- Take the shortest path from v to v'.
- Then $T(v \to v') \leq nd_1 + nd_2 + \cdots = n \cdot \sum_i d_i$.



- Each w outside the path connects to at most 3 path nodes!
- So at most $3n$ edges connect to the path.
- So $\sum_i d_i \leq 5n$ and $T(v \to v') \leq n \cdot 5n = 5n^2$.

Stronger upper bound, cont'd

Recall:

Key Lemma. For any G_n, v, v' we have $T(v \to v') \leq 5n^2$.

Claim [Feige '90]. We have $T(G_n, v) = \mathcal{O}(n^2 \log n)$.

Proof idea:

- Split into phases of $10n^2$ steps each.
- Fix v'. By Lemma and Markov inequality, v' is informed with probability $\geq 1/2$.
- So after k phases, v' is informed unless $1/2^k$.

Stronger upper bound, cont'd

Recall:

Key Lemma. For any G_n, v, v' we have $T(v \to v') \leq 5n^2$.

Claim [Feige '90]. We have $T(G_n, v) = \mathcal{O}(n^2 \log n)$.

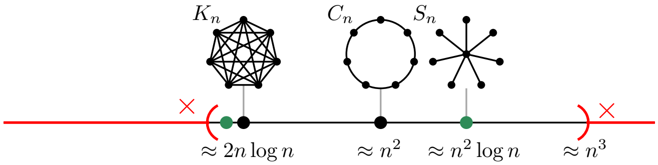
Proof idea:

- Split into phases of $10n^2$ steps each.
- Fix v'. By Lemma and Markov inequality, v' is informed with probability $\geq 1/2$.
- So after k phases, v' is informed unless $1/2^k$.
- By Union bound, k rounds suffice unless $n/2^k$, so

$$T(G_n, v) \leq 10n^2 \cdot \log_2 n + 10n^2 \cdot \sum_{k \geq \log_2 n} \frac{n}{2^k}$$

$$= 10n^2 \log_2 n + \left(1 + \frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \dots\right) = 20n^2 \log_2 n.$$

Landscape



Open Questions:

1. What is the fastest graph?
2. What is the slowest graph? Is it the Star S_n?
3. Efficient algorithm to compute $T(G_n, v)$, for any G_n?

Overview

- PUSH protocol for rumor spreading
- Moran process on graphs

Overview

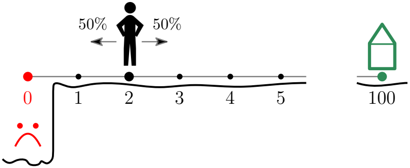
- PUSH protocol for rumor spreading
  - "progressive": only 1 color spreads
- Moran process on graphs

Overview

- PUSH protocol for rumor spreading
  - "progressive": only 1 color spreads
- Moran process on graphs
  - both colors fight!

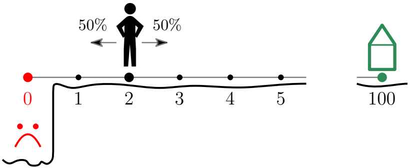
Puzzle: Random Walk

Puzzle. A drunk person, repeatedly stepping left/right 50 : 50.
How likely are they to make it home?



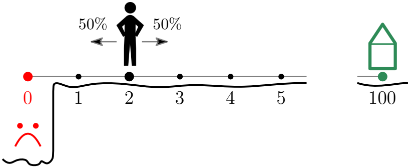
Puzzle: Random Walk

Puzzle. A drunk person, repeatedly stepping left/right 50 : 50.
How likely are they to make it home? Answer: 2%

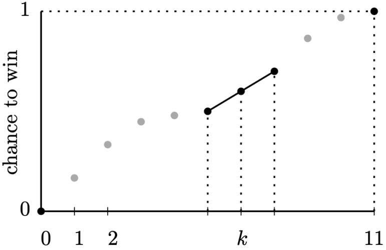


Puzzle: Random Walk

Puzzle. A drunk person, repeatedly stepping left/right 50 : 50.
How likely are they to make it home? Answer: 2%

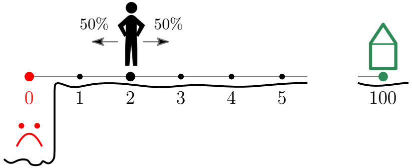


Proof idea: Let p_k = "win-chance from k".
Then $p_k = \frac{1}{2}p_{k-1} + \frac{1}{2}p_{k+1}$.

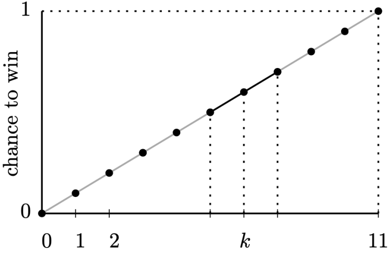


Puzzle: Random Walk

Puzzle. A drunk person, repeatedly stepping left/right 50 : 50.
How likely are they to make it home? Answer: 2%



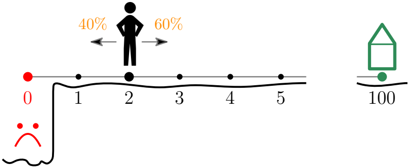
Proof idea: Let $p_k =$"win-chance from k".
Then $p_k = \frac{1}{2}p_{k-1} + \frac{1}{2}p_{k+1}$.



Puzzle: Biased Random Walk

Puzzle. Not completely drunk, steps left/right 40 : 60.

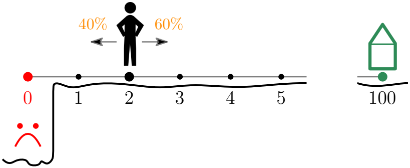
- How likely are they to make it home?



Puzzle: Biased Random Walk

Puzzle. Not completely drunk, steps left/right 40 : 60.

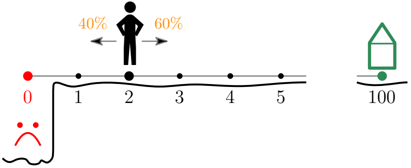
- How likely are they to make it home? $> 55\%$



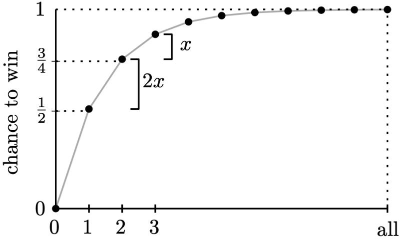
Puzzle: Biased Random Walk

Puzzle. Not completely drunk, steps left/right $40 : 60$.

- How likely are they to make it home? $> 55\%$



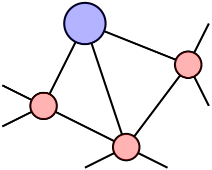
Proof idea: This time $p_k = 0.4 \cdot p_{k-1} + 0.6 \cdot p_{k+1}$.
If it were $\frac{1}{3} : \frac{2}{3}$, the answer would be $\approx 75\%$



Model: Moran process on a graph

[Moran '58] [LHN, '05] A graph $G_n = (V, E)$ on n nodes.

- Nodes: individuals (fitness: residents 1, mutants $r \geq 1$)

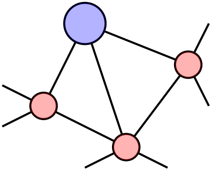


	fitness
mutant	r
resident	1

Model: Moran process on a graph

[Moran '58] [LHN, '05] A graph $G_n = (V, E)$ on n nodes.

- Nodes: individuals (fitness: residents 1, mutants $r \geq 1$)
- Moran Birth-death process on a graph. Repeat:

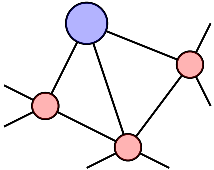


	fitness
mutant	r
resident	1

Model: Moran process on a graph

[Moran '58] [LHN, '05] A graph $G_n = (V, E)$ on n nodes.

- Nodes: individuals (fitness: residents 1, mutants $r \geq 1$)
- Moran Birth-death process on a graph. Repeat:
  1. Birth: Pick a node for reproduction, proportionally to fitness

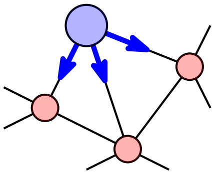


	fitness
mutant	r
resident	1

Model: Moran process on a graph

[Moran '58] [LHN, '05] A graph $G_n = (V, E)$ on n nodes.

- Nodes: individuals (fitness: residents 1, mutants $r \geq 1$)
- Moran Birth-death process on a graph. Repeat:
  1. Birth: Pick a node for reproduction, proportionally to fitness

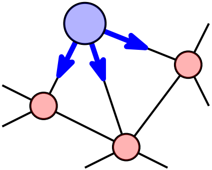


	fitness
mutant	r
resident	1

Model: Moran process on a graph

[Moran '58] [LHN, '05] A graph $G_n = (V, E)$ on n nodes.

- Nodes: individuals (fitness: residents 1, mutants $r \geq 1$)
- Moran Birth-death process on a graph. Repeat:
  1. Birth: Pick a node for reproduction, proportionally to fitness
  2. Death: Pick a neighbor, randomly

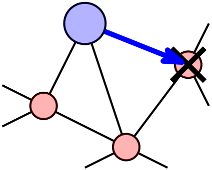


	fitness
mutant	r
resident	1

Model: Moran process on a graph

[Moran '58] [LHN, '05] A graph $G_n = (V, E)$ on n nodes.

- Nodes: individuals (fitness: residents 1, mutants $r \geq 1$)
- Moran Birth-death process on a graph. Repeat:
  1. Birth: Pick a node for reproduction, proportionally to fitness
  2. Death: Pick a neighbor, randomly

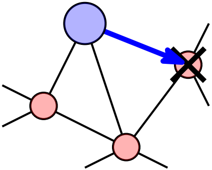


	fitness
mutant	r
resident	1

Model: Moran process on a graph

[Moran '58] [LHN, '05] A graph $G_n = (V, E)$ on n nodes.

- Nodes: individuals (fitness: residents 1, mutants $r \geq 1$)
- Moran Birth-death process on a graph. Repeat:
  1. Birth: Pick a node for reproduction, proportionally to fitness
  2. Death: Pick a neighbor, randomly
  3. Replace

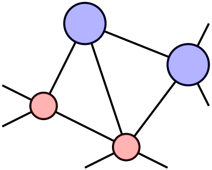


	fitness
mutant	r
resident	1

Model: Moran process on a graph

[Moran '58] [LHN, '05] A graph $G_n = (V, E)$ on n nodes.

- Nodes: individuals (fitness: residents 1, mutants $r \geq 1$)
- Moran Birth-death process on a graph. Repeat:
  1. Birth: Pick a node for reproduction, proportionally to fitness
  2. Death: Pick a neighbor, randomly
  3. Replace

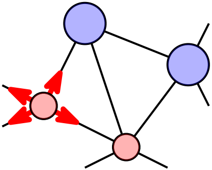


	fitness
mutant	r
resident	1

Model: Moran process on a graph

[Moran '58] [LHN, '05] A graph $G_n = (V, E)$ on n nodes.

- Nodes: individuals (fitness: residents 1, mutants $r \geq 1$)
- Moran Birth-death process on a graph. Repeat:
  1. Birth: Pick a node for reproduction, proportionally to fitness
  2. Death: Pick a neighbor, randomly
  3. Replace

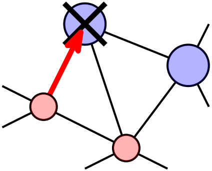


	fitness
mutant	r
resident	1

Model: Moran process on a graph

[Moran '58] [LHN, '05] A graph $G_n = (V, E)$ on n nodes.

- Nodes: individuals (fitness: residents 1, mutants $r \geq 1$)
- Moran Birth-death process on a graph. Repeat:
  1. Birth: Pick a node for reproduction, proportionally to fitness
  2. Death: Pick a neighbor, randomly
  3. Replace

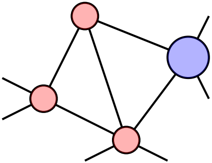


	fitness
mutant	r
resident	1

Model: Moran process on a graph

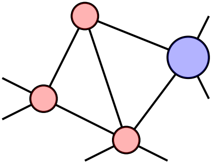
[Moran '58] [LHN, '05] A graph $G_n = (V, E)$ on n nodes.

- Nodes: individuals (fitness: residents 1, mutants $r \geq 1$)
- Moran Birth-death process on a graph. Repeat:
  1. Birth: Pick a node for reproduction, proportionally to fitness
  2. Death: Pick a neighbor, randomly
  3. Replace



	fitness
mutant	r
resident	1

Some features of the Moran process



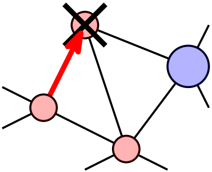
	fitness
mutant	r
resident	1

1. It is stochastic (random).
2. In some steps, nothing happens ("config" stays the same)
3. Nodes can toggle back and forth (more opinions than gossip).
4. Eventually, all nodes become the same type (no mutation).

Quantities of interest:

1. Fixation probability $\mathrm{fp}^r(G_n, v)$: How likely is it that blue wins?
2. Fixation time $\mathrm{T}^r(G_n, v)$: Average #steps until one type wins?
3. ...

Some features of the Moran process



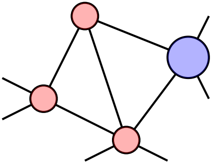
	fitness
mutant	r
resident	1

1. It is stochastic (random).
2. In some steps, nothing happens ("config" stays the same)
3. Nodes can toggle back and forth (more opinions than gossip).
4. Eventually, all nodes become the same type (no mutation).

Quantities of interest:

1. Fixation probability $\mathrm{fp}^r(G_n, v)$: How likely is it that blue wins?
2. Fixation time $\mathrm{T}^r(G_n, v)$: Average #steps until one type wins?
3. ...

Some features of the Moran process



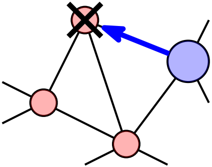
	fitness
mutant	r
resident	1

1. It is stochastic (random).
2. In some steps, nothing happens ("config" stays the same)
3. Nodes can toggle back and forth (more opinions than gossip).
4. Eventually, all nodes become the same type (no mutation).

Quantities of interest:

1. Fixation probability $\mathrm{fp}^r(G_n, v)$: How likely is it that blue wins?
2. Fixation time $\mathrm{T}^r(G_n, v)$: Average #steps until one type wins?
3. . . .

Some features of the Moran process



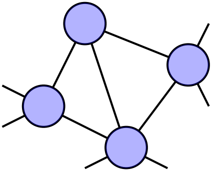
	fitness
mutant	r
resident	1

1. It is stochastic (random).
2. In some steps, nothing happens ("config" stays the same)
3. Nodes can toggle back and forth (more opinions than gossip).
4. Eventually, all nodes become the same type (no mutation).

Quantities of interest:

1. Fixation probability $\mathrm{fp}^r(G_n, v)$: How likely is it that blue wins?
2. Fixation time $\mathrm{T}^r(G_n, v)$: Average #steps until one type wins?
3. ...

Some features of the Moran process



	fitness
mutant	r
resident	1

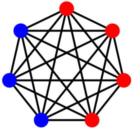
1. It is stochastic (random).
2. In some steps, nothing happens ("config" stays the same)
3. Nodes can toggle back and forth (more opinions than gossip).
4. Eventually, all nodes become the same type (no mutation).

Quantities of interest:

1. Fixation probability $\mathrm{fp}^r(G_n, v)$: How likely is it that blue wins?
2. Fixation time $\mathrm{T}^r(G_n, v)$: Average #steps until one type wins?
3. . . .

Moran process on K_n ($r > 1$)

Idea: Track #mutants and instead of p_k, have p_k^+ and p_k^-.

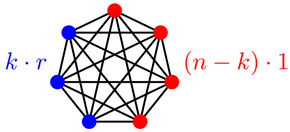


$$F = kr + (n-k)$$

$$\left.\begin{aligned} p_k^+ &= \tfrac{kr}{F} \cdot \tfrac{n-k}{n-1} \\ p_k^- &= \tfrac{n-k}{F} \cdot \tfrac{k}{n-1} \end{aligned}\right\} \quad \frac{p_k^+}{p_k^-} = r$$

Moran process on K_n $(r > 1)$

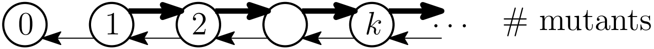
Idea: Track #mutants and instead of p_k, have p_k^+ and p_k^-.



$$F = kr + (n-k)$$

$$\left.\begin{aligned} p_k^+ &= \frac{kr}{F} \cdot \frac{n-k}{n-1} \\ p_k^- &= \frac{n-k}{F} \cdot \frac{k}{n-1} \end{aligned}\right\} \quad \frac{p_k^+}{p_k^-} = r$$

- In each step, we are r-times more likely to gain than to lose a mutant.
- So like the drunk with a sense of direction.

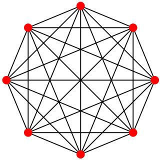


Claim. $\mathrm{fp}^r(K_n) = \frac{1-1/r}{1-1/r^n} \to_{n\to\infty} \boxed{1 - 1/r}$.

Regime $r > 1$ fixed, n large

1. Baseline: Complete graph K_n. We have $\mathrm{fp}^r(K_n) \to 1 - 1/r$.

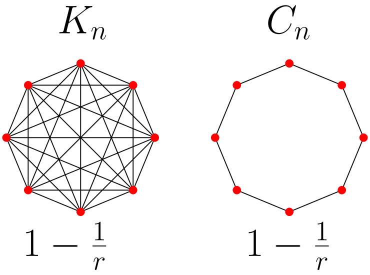
K_n



$1 - \frac{1}{r}$

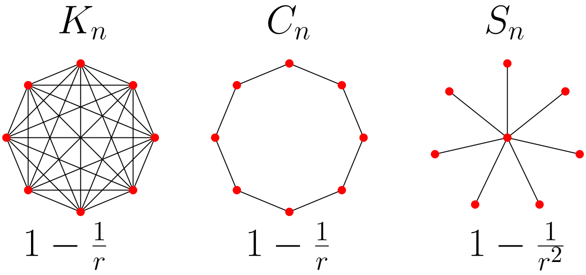
Regime $r > 1$ fixed, n large

1. Baseline: Complete graph K_n. We have $\text{fp}^r(K_n) \to 1 - 1/r$.
2. Isothermal theorem: for any regular graph, e.g. the cycle C_n, we have $\text{fp}^r(C_n) \to 1 - 1/r$.



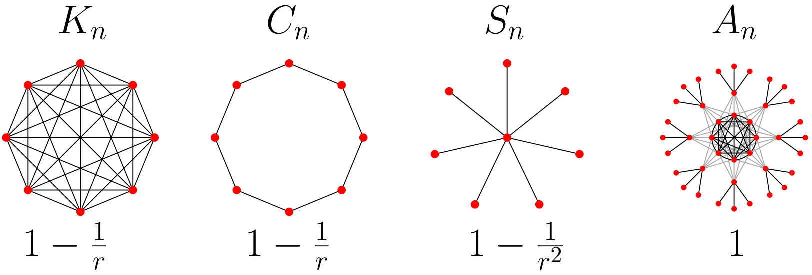
Regime $r > 1$ fixed, n large

1. Baseline: Complete graph K_n. We have $\mathrm{fp}^r(K_n) \to 1 - 1/r$.
2. Isothermal theorem: for any regular graph, e.g. the cycle C_n, we have $\mathrm{fp}^r(C_n) \to 1 - 1/r$.
3. For the star graph S_n we have $\mathrm{fp}^r(S_n) \to 1 - 1/r^2 > 1 - 1/r$.
   - So we would call the star an amplifier.



Regime $r > 1$ fixed, n large

1. Baseline: Complete graph K_n. We have $\text{fp}^r(K_n) \to 1 - 1/r$.
2. Isothermal theorem: for any regular graph, e.g. the cycle C_n, we have $\text{fp}^r(C_n) \to 1 - 1/r$.
3. For the star graph S_n we have $\text{fp}^r(S_n) \to 1 - 1/r^2 > 1 - 1/r$.
   - So we would call the star an amplifier.
4. There exist strong amplifiers: graphs A_n with $\text{fp}^r(A_n) \to_{n\to\infty} 1$, for any $r > 1$.



Finding $\mathrm{fp}^r(G_n, v)$

Finding $\mathrm{fp}^r(G_n, v)$

Computing exactly: Needs solving a linear system of 2^n equations.

Finding $\mathrm{fp}^r(G_n, v)$

Computing exactly: Needs solving a linear system of 2^n equations.

Idea: Approximate, by simulating many times.

Finding $\mathrm{fp}^r(G_n, v)$

Computing exactly: Needs solving a linear system of 2^n equations.

Idea: Approximate, by simulating many times.

- How many times?

Finding $\text{fp}^r(G_n, v)$

Computing exactly: Needs solving a linear system of 2^n equations.

Idea: Approximate, by simulating many times.

- How many times?
- For how long each time?

Finding $\mathrm{fp}^r(G_n, v)$

Computing exactly: Needs solving a linear system of 2^n equations.

Idea: Approximate, by simulating many times.

- How many times?
  - This is OK (for a given precision & confidence).
- For how long each time?

Finding $\mathrm{fp}^r(G_n, v)$

Computing exactly: Needs solving a linear system of 2^n equations.

Idea: Approximate, by simulating many times.

- How many times?
  - This is OK (for a given precision & confidence).
- For how long each time?
  - Do we know anything about $\mathrm{T}^r(G_n, v)$? Yes!

Puzzle: Getting to 100

Puzzle. Roughly how many times do you need to roll a die, on average, until the total becomes 100 or more?

Puzzle: Getting to 100

Puzzle. Roughly how many times do you need to roll a die, on average, until the total becomes 100 or more?

Guess!

Puzzle: Getting to 100

Puzzle. Roughly how many times do you need to roll a die, on average, until the total becomes 100 or more?

Guess!

Attempt 1:

Puzzle: Getting to 100

Puzzle. Roughly how many times do you need to roll a die, on average, until the total becomes 100 or more?

Guess!

Attempt 1: Worst case is 100, best case is $\lceil 100/6 \rceil = 17$, so the average case is $\frac{1}{2}(100 + 17) = 58.5$.

Puzzle: Getting to 100

Puzzle. Roughly how many times do you need to roll a die, on average, until the total becomes 100 or more?

Guess!

Attempt 1: Worst case is 100, best case is $\lceil 100/6 \rceil = 17$, so the average case is $\frac{1}{2}(100 + 17) = 58.5$.

Attempt 2:

Puzzle: Getting to 100

Puzzle. Roughly how many times do you need to roll a die, on average, until the total becomes 100 or more?

Guess!

Attempt 1: Worst case is 100, best case is $\lceil 100/6 \rceil = 17$, so the average case is $\frac{1}{2}(100 + 17) = 58.5$.

Attempt 2: On average, you add $\frac{1}{6} \cdot (1 + 2 + \cdots + 6) = 3.5$ to your total. So you need roughly $100/3.5 = \frac{200}{7} = 28.\overline{571428}$ rolls.

Puzzle: Getting to 100

Puzzle. Roughly how many times do you need to roll a die, on average, until the total becomes 100 or more?

Guess!

Attempt 1: Worst case is 100, best case is $\lceil 100/6 \rceil = 17$, so the average case is $\frac{1}{2}(100 + 17) = 58.5$.

Attempt 2: On average, you add $\frac{1}{6} \cdot (1 + 2 + \cdots + 6) = 3.5$ to your total. So you need roughly $100/3.5 = \frac{200}{7} = 28.\overline{571428}$ rolls.

Solution:

Puzzle: Getting to 100

Puzzle. Roughly how many times do you need to roll a die, on average, until the total becomes 100 or more?

Guess!

Attempt 1: Worst case is 100, best case is $\lceil 100/6 \rceil = 17$, so the average case is $\frac{1}{2}(100 + 17) = 58.5$.

Attempt 2: On average, you add $\frac{1}{6} \cdot (1 + 2 + \cdots + 6) = 3.5$ to your total. So you need roughly $100/3.5 = \frac{200}{7} = 28.\overline{571428}$ rolls.

Solution: But sometimes you overshoot, so the truth is $\doteq 29.04\ldots$

Puzzle: Getting to 100

Puzzle. Roughly how many times do you need to roll a die, on average, until the total becomes 100 or more?

Guess!

Attempt 1: Worst case is 100, best case is $\lceil 100/6 \rceil = 17$, so the average case is $\frac{1}{2}(100 + 17) = 58.5$.

Attempt 2: On average, you add $\frac{1}{6} \cdot (1 + 2 + \cdots + 6) = 3.5$ to your total. So you need roughly $100/3.5 = \frac{200}{7} = 28.\overline{571428}$ rolls.

Solution: But sometimes you overshoot, so the truth is $\doteq 29.04\ldots$

Fact: If you track a quantity $\phi \in [0, B]$ that

- has initial value ϕ_0,
- in each step on average it increases by δ, and
- you stop once it reaches B,

then the expected number of steps until we stop is...

Puzzle: Getting to 100

Puzzle. Roughly how many times do you need to roll a die, on average, until the total becomes 100 or more?

Guess!

Attempt 1: Worst case is 100, best case is $\lceil 100/6 \rceil = 17$, so the average case is $\frac{1}{2}(100 + 17) = 58.5$.

Attempt 2: On average, you add $\frac{1}{6} \cdot (1 + 2 + \cdots + 6) = 3.5$ to your total. So you need roughly $100/3.5 = \frac{200}{7} = 28.\overline{571428}$ rolls.

Solution: But sometimes you overshoot, so the truth is $\doteq 29.04\ldots$

Fact: If you track a quantity $\phi \in [0, B]$ that

- has initial value ϕ_0,
- in each step on average it increases by δ, and
- you stop once it reaches B,

then the expected number of steps until we stop is...

Answer:

Puzzle: Getting to 100

Puzzle. Roughly how many times do you need to roll a die, on average, until the total becomes 100 or more?

Guess!

Attempt 1: Worst case is 100, best case is $\lceil 100/6 \rceil = 17$, so the average case is $\frac{1}{2}(100 + 17) = 58.5$.

Attempt 2: On average, you add $\frac{1}{6} \cdot (1 + 2 + \cdots + 6) = 3.5$ to your total. So you need roughly $100/3.5 = \frac{200}{7} = 28.\overline{571428}$ rolls.

Solution: But sometimes you overshoot, so the truth is $\doteq 29.04\ldots$

Fact: If you track a quantity $\phi \in [0, B]$ that

- has initial value ϕ_0,
- in each step on average it increases by δ, and
- you stop once it reaches B,

then the expected number of steps until we stop is...

Answer: $(B - \phi_0)/\delta$

Puzzle: Getting to 100

Puzzle. Roughly how many times do you need to roll a die, on average, until the total becomes 100 or more?

Guess!

Attempt 1: Worst case is 100, best case is $\lceil 100/6 \rceil = 17$, so the average case is $\frac{1}{2}(100 + 17) = 58.5$.

Attempt 2: On average, you add $\frac{1}{6} \cdot (1 + 2 + \cdots + 6) = 3.5$ to your total. So you need roughly $100/3.5 = \frac{200}{7} = 28.\overline{571428}$ rolls.

Solution: But sometimes you overshoot, so the truth is $\doteq 29.04\ldots$

Fact: If you track a quantity $\phi \in [0, B]$ that

- has initial value ϕ_0,
- in each step on average it increases by δ, and
- you stop once it reaches B,

then the expected number of steps until we stop is...

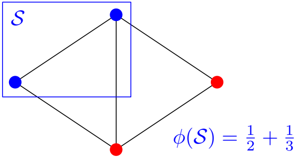
Answer: $(B - \phi_0)/\delta$ ... under mild assumptions.

FPRAS for undirected graphs

Theorem [DGMRSS, '14]. Fix $r \geq 1$. Then $\mathrm{T}^r(G_n, r) = \mathcal{O}(n^4)$.

FPRAS for undirected graphs

Theorem [DGMRSS, '14]. Fix $r \geq 1$. Then $\mathrm{T}^r(G_n, r) = \mathcal{O}(n^4)$.

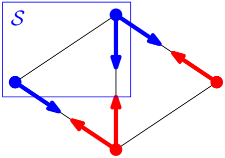


Proof idea:

1. Consider a "potential" $\phi(\mathcal{S}) = \sum_{v \in \mathcal{S}} 1/\deg(v)$. How does it change as we run the process?

FPRAS for undirected graphs

Theorem [DGMRSS, '14]. Fix $r \geq 1$. Then $\mathrm{T}^r(G_n, r) = \mathcal{O}(n^4)$.

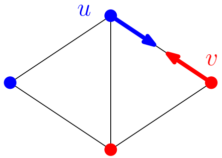


Proof idea:

1. Consider a "potential" $\phi(\mathcal{S}) = \sum_{v \in \mathcal{S}} 1/\deg(v)$. How does it change as we run the process?
2. Claim: contribution of any fixed "active" edge (u, v) is $+\Omega(1/n^3)$.

FPRAS for undirected graphs

Theorem [DGMRSS, '14]. Fix $r \geq 1$. Then $\mathrm{T}^r(G_n, r) = \mathcal{O}(n^4)$.

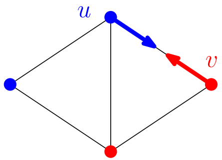


Proof idea:

1. Consider a "potential" $\phi(\mathcal{S}) = \sum_{v \in \mathcal{S}} 1/\deg(v)$. How does it change as we run the process?
2. Claim: contribution of any fixed "active" edge (u, v) is $+\Omega(1/n^3)$.

FPRAS for undirected graphs

Theorem [DGMRSS, '14]. Fix $r \geq 1$. Then $\mathsf{T}^r(G_n, r) = \mathcal{O}(n^4)$.



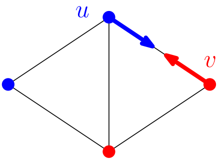
$$\left.\begin{array}{l} p^+ = \frac{r}{F} \cdot \frac{1}{\deg(u)} : \quad +1/\deg(v) \\ p^- = \frac{1}{F} \cdot \frac{1}{\deg(v)} : \quad -1/\deg(u) \end{array}\right\} \quad \Delta_{(uv)} = \frac{r-1}{F} \cdot \frac{1}{\deg(u)\deg(v)}$$

Proof idea:

1. Consider a "potential" $\phi(\mathcal{S}) = \sum_{v \in \mathcal{S}} 1/\deg(v)$. How does it change as we run the process?
2. Claim: contribution of any fixed "active" edge (u, v) is $+\Omega(1/n^3)$.

FPRAS for undirected graphs

Theorem [DGMRSS, '14]. Fix $r \geq 1$. Then $\mathrm{T}^r(G_n, r) = \mathcal{O}(n^4)$.



$$\left.\begin{array}{l} p^+ = \frac{r}{F} \cdot \frac{1}{\deg(u)} : \quad +1/\deg(v) \\ p^- = \frac{1}{F} \cdot \frac{1}{\deg(v)} : \quad -1/\deg(u) \end{array}\right\} \quad \Delta_{(uv)} = \frac{r-1}{F} \cdot \frac{1}{\deg(u)\deg(v)}$$

Proof idea:

1. Consider a "potential" $\phi(\mathcal{S}) = \sum_{v \in \mathcal{S}} 1/\deg(v)$. How does it change as we run the process?
2. Claim: contribution of any fixed "active" edge (u, v) is $+\Omega(1/n^3)$.
3. But $\phi(\mathcal{S}) \in [0, n]$, so ϕ "can't keep increasing for too long".
4. $\Rightarrow T^r(G_n) = \mathcal{O}(n^4)$.

Summary

- Fixation probability can be approximated by simulations.
- Fixation time is between $\Omega(n \log n)$ and $\mathcal{O}(n^{3+\varepsilon})$ [GLR '19].

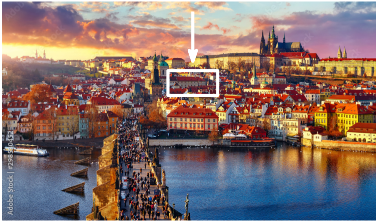
Open questions:

1. Computing $\mathrm{fp}^r(G_n, v)$ exactly?
2. Approximations on directed graphs?
3. Adversarial amplifier?
4. Location-dependent fitness?
5. #mutants at time T?
6. More colors?
7. ...

Wrap-up

- How does stuff spread through networks?
- Even for the simplest model, basic questions are open.
- Puzzles: Jug lemma, Coupon collector, (biased) Random walk

EGT-Prague



Computer Science Institute, Charles Uni

Thank you