MPRI 2.18.1 (2019/20): Distributed algorithms for networks, 2nd part

Lecture 1: Introduction to Gossip-based and Epidemic Algorithms

Surprising Performance with Trivial Algorithms

Benjamin Doerr, LIX

Contents

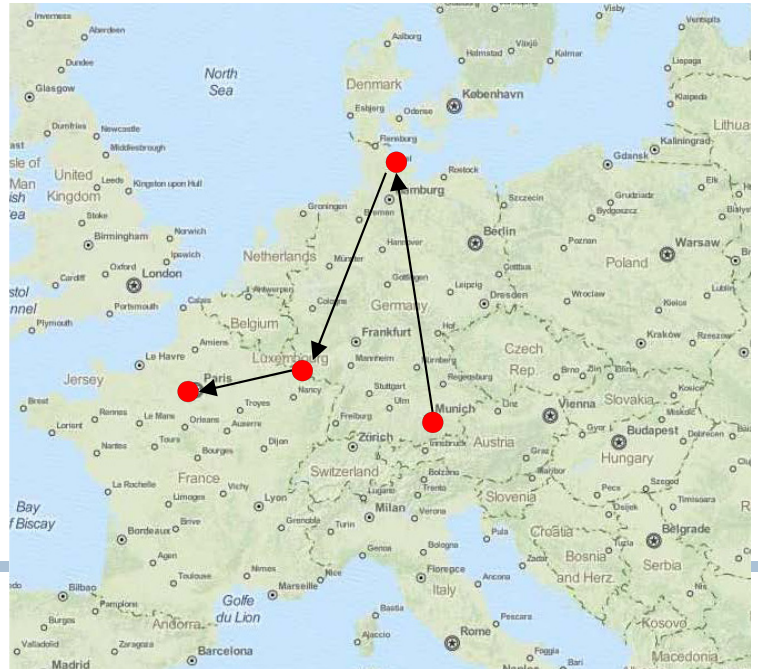
- Course organization
- Introduction epidemic & gossip-based algorithms
- Different phone chains
- Randomized rumor spreading: definition and some results
- Proof of a relatively precise result for complete graphs (analysis of phase 2 deferred to next lecture)
- Methods: birthday paradox, coupon collector, Markov's inequality
- Conclusion
- Appendix: Big-Oh notation
- Homework

CV: Benjamin Doerr

- born in Munich
- Diploma 1998 (Kiel): Group theory
- PhD 2000 (Kiel): Discrete mathematics
- Habilitation 2005 (Kiel): Algorithmic mathematics

- 2005-2013: Senior researcher (≈ directeur de recherche at CNRS), Max Planck Institute for Informatics, Saarbrücken
- since Oct. 2013: Full professor, LIX, École Polytechnique

- research area: randomized methods in computer science (evolutionary algorithms, epidemic algorithms, games, …)



Research Area: Randomized Methods in Computer Science

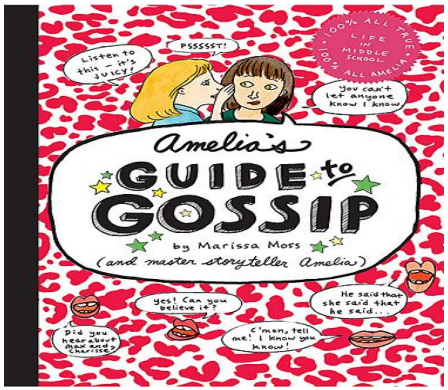
ÉCOLE POLYTECHNIQUE

Randomized Algorithms, esp. Randomized Rounding

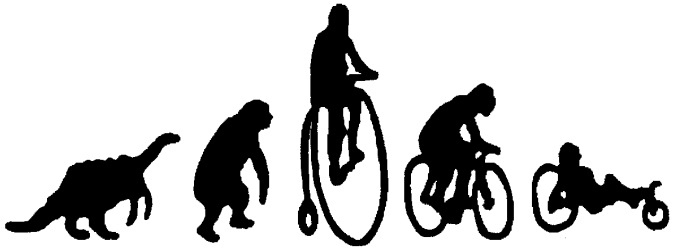
```
int getRandomNumber()
{
    return 4;  // chosen by fair dice roll.
               // guaranteed to be random.
}
```

http://xkcd.com

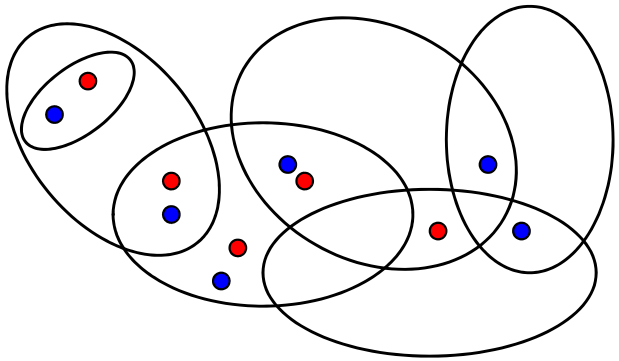
Epidemic Algorithms & Rumor Spreading



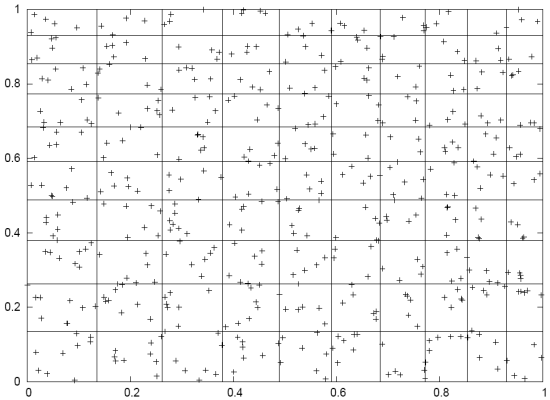
Evolutionary Algorithms & Randomized Search Heuristics



Discrete Maths, Hypergraphs



Discrepancy Theory, Uniform Distribution



Combinatorial Games: Liar Games, Guessing Games, …

Topic of the 2nd Part of the Course: Gossip-Based & Epidemic Algorithms

ÉCOLE POLYTECHNIQUE

- *Gossip-based algorithms ≈ epidemic algorithms*:
  - particular class of distributed algorithms

- *gossip-based:* the main form of communication is that nodes call random neighbors
  - highly robust
  - works well even when topology of network is unknown, changing, …
    → applications: wireless sensor networks, mobile ad-hoc networks

- *epidemic:* information or activity spreads in an epidemic-like manner
  - very fast and efficient
  - strong connections between epidemic algorithms and other epidemic processes (spread of rumors, diseases or malware; forming of opinions)

Examples

- Epidemic algorithms:
  - Communicating updates in replicated database systems: Nodes of the network regularly call random others and exchange updates.
  - Communication in wireless sensor networks: Since the network topology is unknown and unreliable, the only way to send data from A to B is to send it to random neighbors and make them do the same.
  - Peer-to-peer exchange of large data sets: Gain fairness by choosing random paths

- Epidemic processes:
  - How do opinions form in (electronic) social networks, viral marketing
  - Mathematical epidemiology: Model the spread of diseases and analyze possible counter measures
  - Malware/viruses spreading in computer networks

Role in This Course: An Example for a Young Hot Topic

ÉCOLE POLYTECHNIQUE

- *Epidemic algorithms* is a \hot, young, and specialized topic in distributed computing. We discuss such a topic in this course, because
  - it is interesting,
  - it shows to you how emerging research topics look like.

- But: Within this specialized topic, we (intentionally) touch many topics of general interest in distributed algorithms and beyond
  - design and analysis of *randomized algorithms*
  - *random graph* models for wireless sensor networks & social networks

Course Organization

- Teaching days: today (Dec. 12), Dec. 19, Jan. 9, Jan. 16
  - short exam after that (details to be announced)
- Teaching with slides and black-board. I'll update the slides after the course and send them to you. For the black-board parts you should take notes.
- There will be (short) exercises during the course and longer ones between the lectures. This is a chance for you to
  - train your problem-solving skills
  - see how much you understood
  - prepare for the exam
- I like a lot questions and discussions during the lecture ☺, so don't be shy to ask or comment!

Topic Today: Power of Epidemics and Gossiping

ÉCOLE POLYTECHNIQUE

- Toy example *phone chain* showing
  - the power of epidemics
  - the power of gossiping
- Definition: Randomized rumor spreading, the most basic (but central) epidemic/gossip-based algorithm
  - some known results
- Rigorous runtime analysis of randomized rumor spreading in complete networks

Toy Example: Phone Chains

ÉCOLE POLYTECHNIQUE

- Phone chain problem:
  - group of n people
  - one of them knows a piece of information (“rumor”)
  - Task: Communicate this rumor to all others in a collaborative manner!

- In computer science language: devise a good protocol that communicates a piece of information from one node of a network to all nodes?

Traditional Phone Chain

- Traditional phone chain:
  - the set of people has a cyclic order.
  - when you are called (or you are the initially informed person), then you call your successor in the cyclic order

- Properties:
  - work-efficient: $n-1$ calls in total
    - best possible when $n-1$ people are to become informed
  - fair: everyone performs and receives at most one call
  - very slow: $n-1$ rounds of communication
  - not robust: if one call fails, then all subsequent people remain uninformed

Hypercube Phone Chain

- Hypercube phone chain:
  - assume that we have $n = 2^d$ people numbered from 0 to $2^d - 1$
    → think of bit-strings of length d
  - in round $i \in [1..d]$, each informed node x calls node $x\ XOR\ 2^{i-1}$, that is, the bit-string obtained from flipping the $(i-1)$st bit

- Properties:
  - work-efficient: $n - 1$ calls in total
  - relatively fair: nodes perform between 1 and $d = \log_2 n$ calls
  - very fast: $d = \log_2 n$ rounds of communication (this is best possible, since the number of informed nodes can at most double in each round)
    - → "the power of epidemics": making everyone join the process gives a great speed-up!
  - not very robust: if a node stops working in round i, then 2^{d-i} nodes remain uninformed

Randomized (Gossip-based) Phone Chain

- Randomized phone chain:
  - in each round, each node that knows the rumor calls a random node

- First observation: It is not clear when the protocol ends. This is a non-trivial problem, but can be solved. For the moment, let us ignore this and only measure the performance up to the point when everyone is informed.

- Properties (to be proven later):
  - moderately work-efficient: $\approx n \ln n$ calls until everyone is informed
  - relatively fair: $\Theta(\log n)$ calls per node until all informed
  - quite fast: With probability $1 - o(1)$, after $(1 + o(1))(\log_2 n + \ln n)$ rounds of communication everyone is informed
  - very robust: if each call gets lost with 50% chance (unnoticed by the caller), the time increases only by roughly 82%
    - → the power of gossiping: get robustness (almost) for free!

Topic Today: Power of Epidemics and Gossiping

ÉCOLE POLYTECHNIQUE

- [DONE] Toy example *phone chain* showing
  - the power of epidemics
  - the power of gossiping
  - Insight: Calling random neighbors gives efficiency and robustness
- [NEXT] Definition: Randomized rumor spreading, the most basic (but central) epidemic/gossip-based algorithm
  - some known results
- Rigorous runtime analysis of randomized rumor spreading in complete networks

A Fundamental Epidemic Algorithm: Randomized Rumor Spreading (RRS)

ÉCOLE POLYTECHNIQUE

- The randomized phone chain is a communication protocol / epidemic algorithm / gossip-based algorithm that can be implemented in any connected network: *randomized rumor spreading*
  - a round-based process/protocol/algorithm
  - starts with one node informed
  - in each round, each informed node calls a random *neighbor* – this becomes informed if it was not already

- Protocol assumptions:
  - the nodes have some method of synchronization ("rounds")
  - we assume that nodes can call neighbors chosen uniformly at random
  - no interference or other problems if a node is called by several nodes in one round (since we have only one rumor and only informed node call, we can simply assume that it accept an arbitrary incoming call)

Some Known Results

- RRS is fast: $\Theta(\log n)$ rounds suffice to inform with probability at least $1 - 1/n$ all n nodes of…
  - complete graphs [FriezeGrimmet85]
  - hypercubes [FeigePelegRaghavanUpfal90]
  - random graphs $G_{n,p}$, $p \geq (1+\varepsilon)(\log n)/n$ [FeigePRU90]
  - random regular graphs [FountoulakisPanagiotou13]

- RRS is robust: If calls fail independently with probability p…
  - then for any graph the rumor spreading time (roughly) increases at most by a factor of $6/(1-p)$ [ElsässerSauerwald09]
  - in complete graphs: time increases by a factor less than $1/(1-p)$.
    - e.g., by 1.82 when calls fail with probability ½ [DoerrHuberLevavi13]

Plan for the Remaining Time Today

- Obtain a deeper understanding of the working principles of RRS…
- … by trying to prove that RRS in complete networks (= the randomized phone chain) with probability $1 - o(1)$ informs all nodes in $(1 + o(1))(\log_2 n + \ln n)$ rounds
- Side-effect: We learn (revive) some elementary probabilistic arguments that are very useful for many computer science problems

Plan for the Remaining Time

- Your task now (5-10 minutes individual work), after that our joint task: Try to prove that with probability $1 - o(1)$, RRS informs all nodes within $(1 + o(1))(\log_2 n + \ln n)$ rounds.
  - First try to get a rough (informal!) understanding of what is going on.
  - Only then try to find rigorous arguments.

- First hint: Perfect symmetry and the rounds are independent
  - we don't care which nodes are informed, only their number is relevant
  - we can ignore how these nodes became informed
  - → arguments of type "if we start a round with k informed nodes, then we end this round with … informed nodes" could make sense

- Second hint: Things are randomized, so we need to understand the typical behavior (high probability, expectations)

Good luck!

Overview: RRS in Complete Graphs

ÉCOLE POLYTECHNIQUE

The rumor spreading process can be split into 4 phases:

- 1st phase (up to $o(\sqrt{n})$ informed nodes): True doubling. With high probability, all calls in one round reach a "new" node, that is, the number of informed nodes doubles in these rounds. → "Birthday paradox."
- 2nd phase (up to $o(n)$ informed nodes): Exponential growth. Each call still has a probability of $1 - o(1)$ of reaching a "new" node. Hence the number of informed nodes almost doubles, that is, increases by a factor of $(2 - o(1))$ in these phases
- 3rd phase: a short connecting phase between the exponential growth of the informed nodes and the exponential shrinking of the uninformed nodes
- 4th phase (from $n - o(n)$ informed nodes on): Exponential shrinking of the uninformed nodes by a factor of $(e - o(1))$. → "Coupon collector"

Preliminaries

- Perfect symmetry, rounds are “independent” (Markov process)
  - we don’t care which nodes are informed, only their number is relevant
  - we can ignore how these nodes became informed
  - → arguments of type “if we start a round with k informed nodes, then we end this round with … informed nodes” make sense
- Assumption: Nodes call a random node including themselves:
  - only slows down things (but not significantly)
  - adds symmetry
  - eases writing: “$1/n$” instead of “$1/(n-1)$”
- Write $\log n := \log_2 n$
- Plan: Show that each phase does what we want with probability $1 - o(1)$
  - for didactic reasons: analyze 1st , 4th, 3rd, and then 2nd phase.

Phase 1: True Doubling

- Lemma: If $k = o(\sqrt{n})$, then with probability $1 - o(1)$ the first k calls of the process (no matter in how many rounds they take place) all reach a new node.

- Proof 1: Simply compute the probability.
  - $(1 - 1/n) \cdot (1 - 2/n) \cdot \ \ldots \cdot (1 - k/n) \geq 1 - \sum_{i=1}^{k} i/n = 1 - \frac{(k+1)k}{2n}$

general version of Bernoulli's inequality

- Proof 2: Estimate wisely (recommended).
  - For each of the first k calls, at most k nodes are informed when they take place. Hence each has a "failure" probability of at most k/n.
  - Union bound: Probability that some call fails $\leq$ sum (over all calls) of their failure probability = $k \cdot \ k/n = k^2/n$

- In both cases, the total failure probability is $o(1)$ if $k = o(\sqrt{n})$.

Detailed Version of Proof 1

- Assume that the nodes are numbered from 1 to n.
- Consider the first k calls (it does not matter who is calling, inside the rounds we order the calls arbitrarily). Let $X_1, \dots, X_k$ be the targets of these calls. Let X_0 be the initially informed node.
- The X_i are independent and uniformly distributed in $[1..n]$

- $\Pr[X_0, \dots, X_k \text{ are all different}]$
- $= \Pr[X_1 \neq X_0 \wedge X_2 \notin \{X_0, X_1\} \wedge X_3 \notin \{X_0, X_1, X_2\} \wedge \cdots \wedge X_k \notin \{X_0, \dots, X_{k-1}\}]$
- $= \Pr[X_1 \neq X_0] \Pr[X_2 \notin \{X_0, X_1\} \mid X_1 \neq X_0] \Pr[X_3 \notin \{X_0, \dots, X_2\} \mid X_1 \neq X_0, X_2 \notin \{X_0, X_1\}] \dots$
- $= (1 - 1/n) \cdot (1 - 2/n) \cdot \dots \cdot (1 - k/n) \geq 1 - \sum_{i=1}^{k} i/n$
- $= 1 - k(k+1)/2n$

Excursus: Birthday Paradox

- Imagine the year has n days and you have k people having birthday on a random day. What is the chance that at least two have the same birthday?
  - $o(1)$, if $k = o(\sqrt{n})$
  - $1 - o(1)$, if $k = \omega(\sqrt{n})$
  - Surprisingly high "in practice": 23 people are enough to have a 50% chance for a double-birthday when $n = 365$
- Proof: Estimate the probability of proof 1 on the previous slide
- Note:
  - It is no rocket science to compute these probabilities.
  - But: It is good to have this "rule" that things are decided at $\sqrt{n}$ in the back of your mind (so that you don't have to calculate, but you know immediately what to roughly expect)

True Doubling: Final Proof

- Previous slides: If $k = o(\sqrt{n})$, then with probability $1 - o(1)$ the first k calls of the process (no matter in how many rounds they take place) all reach a new node.

- Corollary (Phase 1): Let $T \leq (0.5 - (\log n)^{-0.5}) \log n$. With probability $1 - o(1)$, after the first T rounds 2^T nodes are informed.

- Proof:
  - Consider the first $k = 2^T - 1 = o(\sqrt{n})$ calls of the process.
  - By the previous lemma, all calls inform a "new" node (with probability $1 - o(1)$).
  - A simple induction shows that the number of informed nodes doubles in each of the first T rounds (in the non-failure case).

Phase 4: Exponential Shrinking

- Lemma: If at some time there are $i = n - o(n)$ nodes informed, then with probability $1 - o(1)$, after $T = (n/i)\ln(n) = \big(1 + o(1)\big)\ln n$ rounds all nodes are informed.

- Proof:
  - The probability that a fixed uniformed node remains uninformed for T rounds is at most

$$\left(1 - \frac{1}{n}\right)^{iT} \leq \exp\left(-\frac{iT}{n}\right) \leq \frac{1}{n}$$

    [note the $1 + r \leq e^r$ argument valid for all real numbers r]
  - Union bound: The probability that one of the $o(n)$ uninformed nodes remains uninformed for T rounds is at most $o(n)(1/n) = o(1)$

- Note: Resembles the *coupon collector problem* (next slide)
  - uninformed nodes are missing coupons, get i coupons per round

Excursus: Coupon Collector

- Coupon collector process:
  - You start with no coupons
  - In each round, you get a coupon having a random type out of n different types
  - Question: How long does it take until you have at least one coupon of each type?

- Theorem: The coupon collector time T satisfies
  - $E[T] = n(1 + 1/2 + 1/3 + \cdots + 1/n) = nH_n = n \ln n + \Theta(n)$
  - $\Pr[T \geq (1+\varepsilon) n \ln n] \leq n^{-\varepsilon}$
  - $\Pr[T < (1-\varepsilon)(n-1) \ln n] \leq \exp(-n^{\varepsilon})$

- Note: another useful building block in randomized algorithms

Summary: Rumor Spreading in Complete Networks

- Split the process into 4 phases that all reach their target with probability at least $1 - o(1)$
  - Phase 1: Perfect doubling up to $o(\sqrt{n})$ informed nodes [birthday par.]
  - Phase 2 [details next lecture]: Almost doubling
    - Phase 1+2: Obtain $i_2 = n/(\log\log n)^2$ informed nodes within $T_{12} = \log n + 2\log\log n$ rounds
  - Phase 3 [see homework]: From i_2 informed nodes to i_2 uninformed nodes in time $T_3 = (\log\log n)^4$
  - Phase 4: Exponential shrinking of uninformed by a factor of at least $e(1 - o(1))$ → $T_4 = \ln(n)(1 + o(1))$ rounds suffice for this phase [coupon collector]

- Result: With prob. $1 - o(1)$, after $T_{12} + T_3 + T_4 = (1 + o(1))(\log n + \ln n)$ rounds all nodes are informed.

Summary

- Epidemic/gossip-based algorithms: The simple idea of talking to random neighbors gives very efficient and robust algorithms
- Randomized rumor spreading informs all nodes of a complete graph on n vertices in $(1 + o(1))(\log_2 n + \ln n)$ rounds (with high probability).
- Simple methods clever applied
  - coupon collector
  - birthday paradox
  - union bound
  - Markov's inequality
  - $1 + x \leq e^x$ for all real numbers x

Appendix: Reminder
Big-Oh Notation (Landau Symbols)

- We frequently use so-called *Landau symbols* (big-Oh notation) to roughly describe runtimes or other quantities. You should known (some of) them from earlier courses, but maybe not all (e.g., $o(.), \omega(.)$, ...).

- Simple definition $O(.)$ (sufficient for most purpose in computer science):
  - Let $X \subseteq \mathbb{R}$ be unbounded (arbitrary large numbers exist in X).
  - Let $f, g: X \to \mathbb{R}$ be real functions.
  - Definition: We write $f = O(g)$ if and only if there are $M, x_0 \in \mathbb{R}$ such that $|f(x)| \leq M|g(x)|$ for all $x \geq x_0$.

- Warning: This "=" is not symmetric. A better notation (used by some, but not many) would be $f \in O(g)$. Then $O(g)$ would be the class of all functions having asymptotically not larger growth than g.

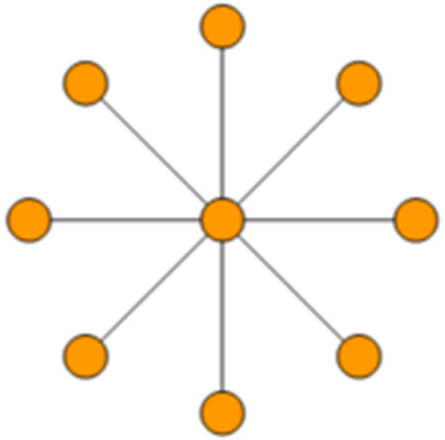
Big-Oh Notation (2)

- Idea: This big-Oh notation makes a statement on the growth behavior of f without caring about constant factors and lower order terms
  - $0.01n^2 = O(n^2)$ and $100n^2 = O(n^2)$
  - $n^2 - 1000n = O(n^2)$ and $n^2 + 1000n = O(n^2)$
- Friends of big-Oh:
  - $f = \Omega(g) :\Leftrightarrow g = O(f)$
  - $f = \Theta(g) :\Leftrightarrow f = O(g) \wedge f = \Omega(g)$
  - $f = o(g) :\Leftrightarrow \forall \varepsilon > 0\ \exists x_0 \in \mathbb{R}\ \forall x \geq x_0 : |f(x)| \leq \varepsilon |g(x)|$
    - equivalent to $\lim_{x \to \infty} f(x)/g(x) = 0$ if $g \neq 0$
  - $f = \omega(g) :\Leftrightarrow g = o(f)$
- Notation: We use this also without writing down g explicitly. E.g.,
  - $f = O(n^2)$, with prob. $1 - o(1)$, …

Homework 1.1: Rumor Spreading in Star Graphs [Easy]

- Let G be the star graph on n vertices, that is, G has n vertices such that there is one that is connected to all other $n-1$ vertices via a direct edge. Apart from this, there are no edges.

- Assume that you run the randomized rumor spreading protocol in this graph: the rumor starts in some node and then in each round, each informed node calls a random neighbor (not itself). For each starting vertex, compute precisely the expected time until all vertices informed.



A star graph on 9 vertices.

Homework 1.2: Phase 3 (Short Connecting Phase) [Easy]

ÉCOLE POLYTECHNIQUE

- Consider randomized rumor spreading in a complete graph on n vertices as discussed in the lecture today. We now want to analyze the short connecting phase between doubling and shrinking.

- Let $f, g = o(1)$. Assume that we have already $i = nf$ nodes informed. Let $T = 1/(fg)$. Show that with probability $1 - o(1)$, after T rounds at least $n(1 - g)$ nodes are informed.

- Hints:
  - Compute the expected number of uninformed nodes after T rounds with arguments similar to the ones we used to analyze Phase 4.
  - Use Markov's inequality to show the claim.

Homework 1.3: Start of the Analysis of Phase 2 [Fairly Easy]

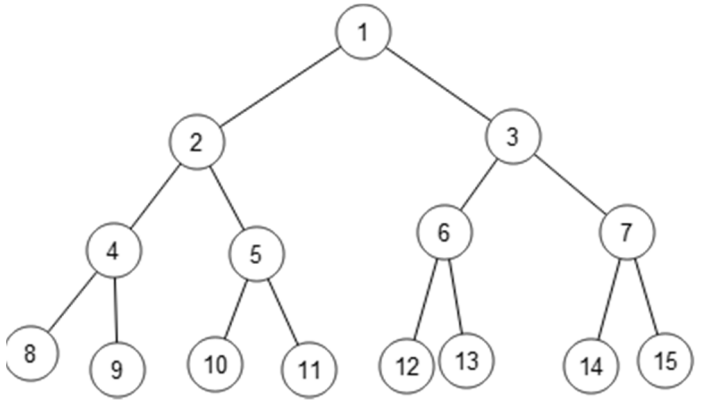
- Consider randomized rumor spreading in a complete graph on n vertices as discussed in the lecture today.

- Assume that a round starts with i nodes informed. Let X denote the number of nodes newly informed in this round (that is, the number of nodes that were not informed at the start of the round, but are informed at the end).

- Prove that $E[X] \geq i\left(1 - \frac{3i}{2n}\right)$!
  - Please try as much as possible to transform your intuitive understanding into a solid proof.
  - Note: If you can only prove $E[X] \geq i\left(1 - \frac{2i}{n}\right)$, that is still good (and will change nothing for the final result)

Homework 1.4: Rumor Spreading in Trees [not so Easy]

- Let G be a k-regular rooted tree of height h, that is, an undirected graph having $n = 1 + k + k^2 + \cdots + k^h$ vertices such that there is one "root" vertex which has k neighbors such that each of them is the root of a k-regular tree of height $h - 1$ (when we delete the original root and all edges incident with it). Assume that you run the randomized rumor spreading protocol in this graph, starting the rumor in the root.



- Task: Analyze how long it takes until all nodes are informed!

- Hint: Proceed as follows.
  - Get a rough informal understanding of what is going on.
  - Prove an upper bound on the rumor spreading time.
  - Complement this with a matching lower bound.

Homework 1.5: Rumors in Arbitrary Graphs [Short, but Tough]

ÉCOLE POLYTECHNIQUE

- Let G be any undirected graph with n vertices. Let x and y be two vertices of G. Assume that you start the randomized rumor spreading process in the vertex x.

- Task: Prove that the expected time it takes until y is informed is at most $3n$.