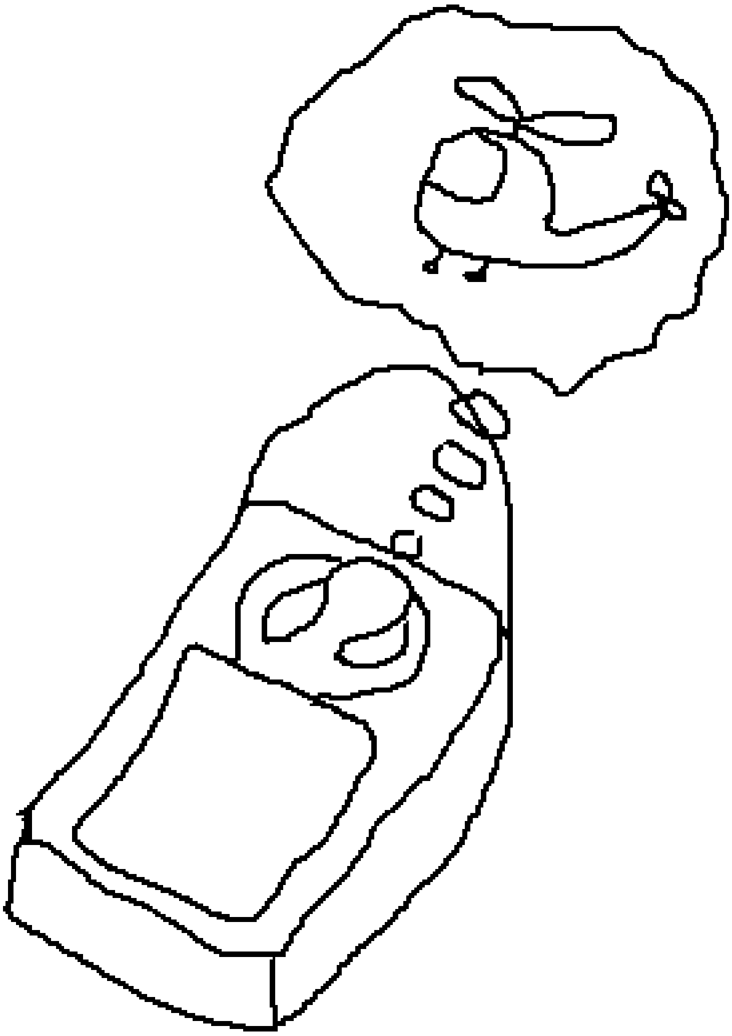
Average-case analysis & algorithms



~~Worst~~ case

- **Problem = (L, P_r)** where P_r is a probability function over the input strings: $P_r : \sum^* \to [0, 1]$.
- $\sum_{x \in \sum^*} P_r(x) = 1$ (the probabilities must sum up to 1).
- **Average time** of an algorithm:

$$T_A(n) = \sum_{\{x \in \sum^* \,\big|\, |x|=n\}} T_A(x) P_r(x)$$

- **Key issue:** How to choose P_r so that it is a realistic model of reality.
- Natural solution: Assume that all instances of length n are equally probable (uniform distribution).

Random graphs

Uniform probability model (UPM)

- Every graph G has equal probability
- If the number of nodes = n, then $P_r(G) = \frac{1}{\#\text{graphs}} = \frac{1}{2^{\binom{n}{2}}}$, where $\binom{n}{2} = \frac{n(n-1)}{2}$
- UPM is more natural for interpretation

Independent edge probability model (IEPM)

- Every possible edge in a graph G has equal probabililty p of occuring
- The edges are independent in the sense that for each pair (s,t) of vertices, we make a new toss with the coin to decide whether there will be an edge between s and t.
- For $p = \frac{1}{2}$ IEPM is identical to UPM:

$$P_r(G) = \left(\frac{1}{2}\right)^m \cdot \left(\frac{1}{2}\right)^{\binom{n}{2}-m} = \frac{1}{2^{\binom{n}{2}}}$$

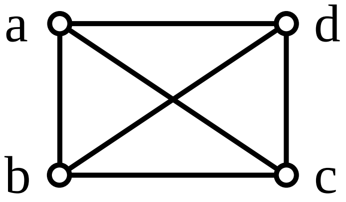
- IEPM is easier to work with

Example: 3-COLORABILITY

In 3-COLORABILITY we are given a graph as input and we are asked to decide whether it is possible to color the nodes using 3 different colors in such a way that any two nodes have different colors if there is an edge between them.

Theorem 1 3-COLORABILITY, *which is an $\mathcal{NP}$-complete problem, is solvable in* ***constant average (expected) time*** *on the IEPM with $p = 1/2$ by a branch-and-bound algorithm (with exponential worst-case complexity).*

Proof:
Strategy (for a rough estimate): Use the indep. edge prob. model. Estimate expected time for finding a proof of non-3-colorability.



K_4 (a clique of size 4) is a proof of non-3-colorability.

- The probability of 4 nodes being a K_4:
  $P_r(K_4) = 2^{-\binom{4}{2}} = 2^{-6} = \frac{1}{128}$
- Expected no. of 4-vertex sets examined before a K_4 is found:

$$\sum_{i=1}^{\infty} i\left(1-2^{-6}\right)^{i-1} 2^{-6} = 2^{-6} \sum_{i=1}^{\infty} i\left(1-2^{-6}\right)^{i-1}$$

$$\stackrel{*}{=} 2^{-6} \frac{1}{\left(1-\left(1-2^{-6}\right)\right)^{2}}$$

$$= 2^{-6} \frac{1}{\left(2^{-6}\right)^{2}} = \frac{2^{12}}{2^{6}} = 2^{6} = 128$$

— $\left(1-2^{-6}\right)^{i-1} 2^{-6}$ is the probability that the first K_4 is found after examining exactly i 4-vertex sets.

— $(*)$ is correct due to the following formula ($q = 1 - 2^{-6}$) from mathematics (MA100):

$$\sum_{i=1}^{\infty} i q^{i-1} = \frac{\delta}{\delta q}\left(\sum_{i=1}^{\infty} q^{i}\right) = \frac{\delta}{\delta q}\left(\frac{q}{1-q}\right)$$

$$= \frac{1}{(1-q)^{2}}$$

Conlusion: Using IEPM with $p = \frac{1}{2}$ we need to check 128 four-vertex sets on average before we find a K_4.

Note: Random graphs with constant edge probability are very dense (have lots of edges). More realistic models has p as a function of n (the number of vertices), i.e. $p = 1/\sqrt{n}$ or $p = 5/n$.

0-1 Laws

as a link between probabilistic and deterministic thinking.

Example: "Almost all" graphs are

- not 3-colorable
- Hamiltonian
- connected
- ...

Def. 1 *A property of graphs or strings or other kind of problem instances is said to have a **zero-one law** if the limit of the probability that a graph/string/problem instance has that property is either* 0 *or* 1 *when* n *tends to infinity* ($\lim_{n\to\infty}$)*.*

Example: HAMILTONICITY

a linear expected-time algorithm for random graphs with $p = 1/2$.

- **Difficulty:** The probability of non-Hamiltonicity is too large to be ignored, e.g. $P_r(\exists$ at least 1 isolated vertex$) = 2^{-n}$.
- The algorithm has 3 phases:
  — **Phase 1:** Construct a Hamiltonian path in linear time. Fails with probability $P_1(n)$.
  — **Phase 2:** Find proof of non-Hamiltonicity or construct Hamiltonian path in time $\mathcal{O}\left(n^2\right)$. Unsuccessful with probability $P_2(n)$.
  — **Phase 3:** Exhaustive search (dynamic programming) in time $\mathcal{O}\left(2^{2n}\right)$.
- Expected running time is
  $\leq \mathcal{O}(n) + \mathcal{O}\left(n^2\right) P_1(n) + \mathcal{O}\left(2^{2n}\right) P_1(n)P_2(n)$

$$= \mathcal{O}(n) \text{ if } P_1(n) \cdot \mathcal{O}\left(n^2\right) = \mathcal{O}(n)$$
$$\text{and } P_1(n)P_2(n) \cdot \mathcal{O}\left(2^{2n}\right) = \mathcal{O}(n)$$

- Phase 2 is necessary because $\mathcal{O}\left(2^{-n}\right) \cdot \mathcal{O}\left(2^{2n}\right) = \mathcal{O}\left(2^n\right)$.
- After failing to construct a Hamiltonian path fast in phase 1, we first reduce the probability of the instance being non-Hamiltonian (phase 2), before doing exhaustive search in phase 3.