Randomized Algorithms

why study randomized algorithms?

Speed: often are faster than deterministic ones even if their worst-case time is worse (e.g. QuickSort)

Simplicity: Easier to implement ($\sim$ faster)

Only thing we know: For many problems (in particular online algorithms) the only way we can beat an adversary is using randomness. Any deterministic algorithm will be bad. or checking if a multivariable polynomial is identical to zero.

Randomized algorithm $\neq$ Average case analysis

In average case analysis we assume some dist. of input (e.g. uniformly random) but when designing randomized algorithms we don't have such assumptions.

Types of randomized algorithms:

- Las Vegas: Always find the correct answer but the running time is a random function that can vary. We compute the expected time.

  e.g. randomized QuickSort, expected time is $\Theta(n \log n)$ but worst case $\Theta(n^2)$.

- Monte Carlo: These algorithms may not find the correct answer always; we show that they find the correct answer with some Probability $0 < P < 1$.

  Note: we can always increase the probability by running multiple independent copies.

  The running time might also be bounded in expectation.

Some Paradigms for Randomized Algorithms

- Avoiding adversarial input: randomly reorder.
- Fingerprinting & hashing: use smaller (randomly generated) fingerprints to compare large objects.

- Random Sampling: used in approximate counting of large sets, median finding, etc.
- Online algorithms / load balancing: spread the load among resources, reduce cost / time
- Symmetry breaking: protocols in network, avoiding deadlock in distributed systems.
- Probabilistic method: Very powerful technique in combinatorics. To show the existence of an object one shows the Probability of a carefully chosen one having the desired property is > 0. (e.g. Lovasz local Lemma)

  There are methods to turn these into Algorithms

Probability Background

Probability Space S: Set of possible outcomes of a (usually random) process

e.g. rolling a die $\{1, 2, \ldots, 6\}$; flipping a coin $\{H, T\}$

Event $\mathcal{E}$: Any subset of the probability space. $\mathcal{E} \subseteq S$

Probability distribution: A func $Pr: S \longrightarrow [0,1]$ s.t.

$$\sum_{e \in S} Pr(e) = 1$$

Random variable X: It is a function from S to reals or integers.

e.g. space of rolling a pair of dice.

define X be the random variable that is 0/1 depending on whether the parity of sum of the two faces is even/odd.

- Expected value: $E[X] = \mu(X) = \sum_x Pr[X=x]$
- Variance: $Var[X] = E[(X-E[X])^2]$

$$E[(X-E[X])^2] = E[X^2] - 2E[X \cdot E[X]] + E[X]^2$$

$$= E[X^2] - E[X]^2$$

- Standard deviation: $\sigma(X) = \sqrt{Var(X)}$
- Rules for $E[\cdot]$ and $Var(\cdot)$:

 * linearty of expectation: $E[\sum_i \alpha_i X_i] = \sum_i \alpha_i E[X_i]$

 if X & Y are independent: $E[X \cdot Y] = E[X] \cdot E[Y]$

Simple Deviation Bounds:

Markov's inequality: Let X be a random

moment

Markov's inequality: Let X be a random variable that takes non-negative values.

first moment method

$$\forall\, t>0: \quad \Pr[X \geq t] \leq \frac{E[X]}{t}.$$

or $$\Pr[X \geq t \cdot E[X]] \leq \frac{1}{t}$$

Proof: Define $I = \begin{cases} 1 & \text{if } X \geq t \\ 0 & \text{o.w.} \end{cases}$

Since $X \geq 0$, $I \leq \frac{X}{t}$.

$$E[I] = \Pr[I=1] = \Pr[X \geq t] \leq E\left[\frac{X}{t}\right] = \frac{1}{t} \cdot E[X]$$

Chebyshev's inequality:

second moment method

For any random variable X and any $\lambda > 0$:

$$\Pr[(X - E[X]) \geq \lambda] \leq \frac{Var[X]}{\lambda^2}$$

or $$\Pr[(X - E[X]) \geq \lambda E[X]] \leq \frac{Var[X]}{\lambda^2 E[X]^2}$$

Proof: $$\Pr[|X - E[X]| \geq \lambda] = \Pr[(X - E[X])^2 \geq \lambda^2]$$

by Markov's $$\leq \frac{E[(X-E[X])^2]}{\lambda^2}$$

$$= \frac{Var[X]}{\lambda^2}.$$

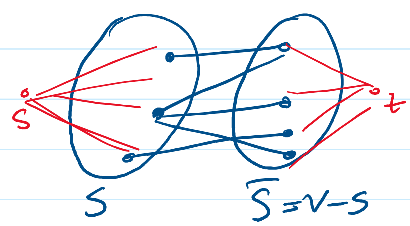
Use of Markov's ineq. is called "first moment" method & using Chebyshev's "second moment."

Karger's Randomized Min_Cut

Definition: Given an undirected graph $G=(V,E)$ a cut is a partition of V into two non-empty set S and $\bar{S}=V-S$. Size of the cut, $\delta(S,\bar{S})$: # of edges between S and $\bar{S}$ (if the graph is weighted, we consider edge weights).

Min_cut Problem: Find a cut $(S,\bar{S})$ with smallest size.

$$\lambda(G) = \min_{S \subsetneq V} \delta(S,\bar{S})$$

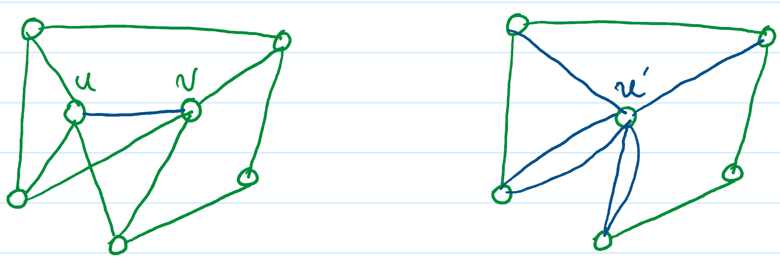


- one can try to use Max_flow_min_cut we can try each pair of vertices s,t as source_Sink and find a min_s,t_cut.
- if $F(m,n)$ is the time complexity of Max_flow in graphs with $|V|=n$, $|E|=m$ this takes $O(n^2.F(m,n))$.
- This can be easily improved to $O(n.F(m,n))$ (just iterate over nodes as sink, how?)

- Time complexity for max-flow has improved over a long line of research to almost linear time $O(m^{1+o(1)})$ [Chen, Kyng, Liu, Yang, Peng, Gutenberg, Sachdeva 2023]

- we present an $\tilde{O}(n^2)$ randomized algorithm (due to D. Karger & C. Stein '93).

Contraction: Recall that contraction of edge $e = uv$ in $G = (V, E)$ gives a multigraph $G' = G/e$ where we contract nodes u, v into a single node in G', delete all edges between u, v



- To represent the multigraph we can keep the multiplicity of edges between two nodes.

Lemma 1: For any edge $e \in G$, any cut in G/e corresponds to a cut of same size in G.

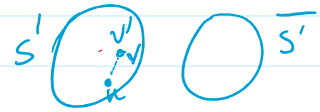
proof: Consider any edge $e = uv \in G$ and suppose it is contracted into v' in G'. Any cut $(S', \bar{S}')$

in G' has v' in one side; WLOG say $v' \in S'$.

consider cut $(S, \bar{S})$ in G where

$S = (S' - v') \cup \{u, v\}$. It is easy to see that

$\delta(S, \bar{S}) = \delta(S', \bar{S}')$.



Corollary 1: if the min_cut of G has size k then min_cut of G/e has size at least k.

– These suggest the following algorithm:

Simple_min_cut (G):

– $G_0 \leftarrow G$ and $i \leftarrow 0$

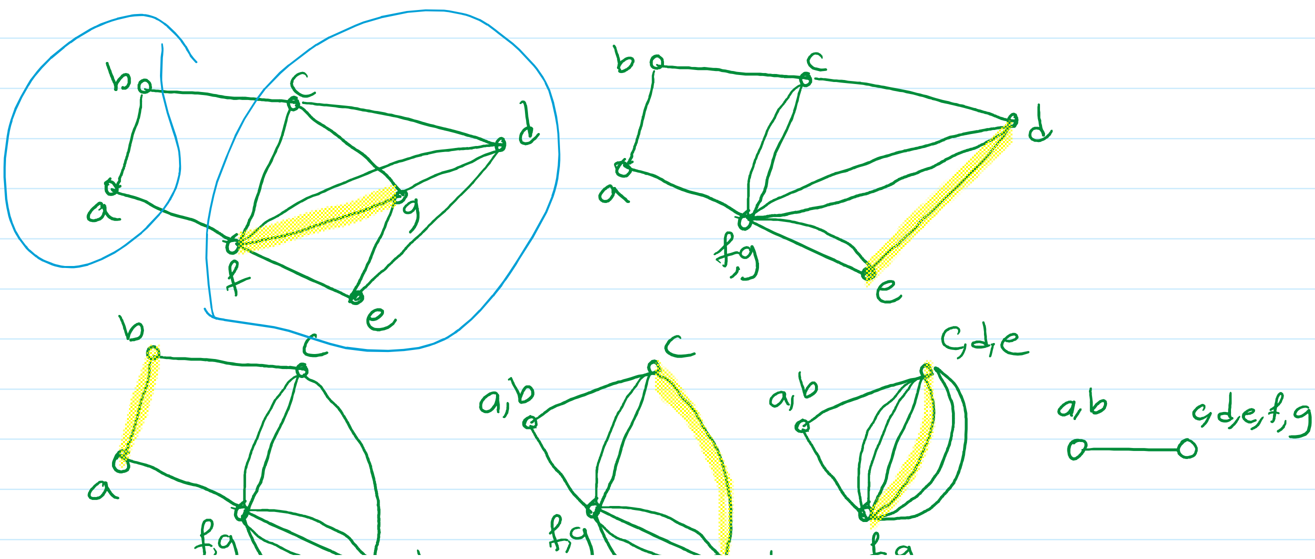
– while G_i has >2 vertices do

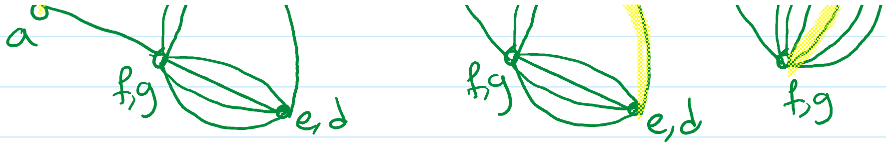
  pick a random edge $e \in G_i$ u.r.

  $G_{i+1} \leftarrow G_i / e$

  $i \leftarrow i+1$

– return edges of G_i





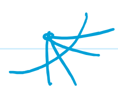
Time complexity: Clearly we have $n-2$ iterations; the contraction step can be done in $O(n)$ as there are at most $O(n)$ edges that change. $\longrightarrow$ total time $O(n^2)$

– the above lemma applied inductively and the corollary imply:

Corollary 2: $\forall$ cut in G_i $(0 \le i \le n-1)$ corresponds to a cut of same size in G. Also $\lambda_{G_i} \geq \lambda_G$

Fact: $\forall$ graph G, if $\lambda_G = k$ then $m \geq \frac{kn}{2}$

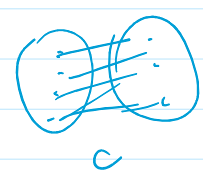
proof: Easy



$\deg(v) \geq k$

Corollary 3: $|E(G_i)| \geq \frac{(n-i)k}{2}$

– suppose $C \subseteq E$ is a fixed min-cut of G. Clearly the probability that the algorithm returns C is the same probability that none of edges of C are contracted. let $|C| = k$.



– let $\mathcal{E}_i$ be the event that we didn't pick any edge of C in iteration i. Then:

Since G_i has $\geq \frac{nk}{2}$ edges

$$\Pr[\mathcal{E}_1] \geq 1 - \frac{k}{nk/2} = 1 - \frac{2}{n}$$

$$\Pr\left[\mathcal{E}_i \,\middle|\, \bigcap_{j=1}^{i-1} \mathcal{E}_j\right] \geq 1 - \frac{2}{n-i+1}$$

Since the events (iterations) are independent, the probability of event $\mathcal{E}$ (no edge of C ever selected)

$$\Pr[\mathcal{E}] = \prod_{i=1}^{n-2} \Pr\left[\mathcal{E}_i \,\middle|\, \bigcap_{j=1}^{i-1} \mathcal{E}_j\right]$$

$$\geq \prod_{i=1}^{n-2} \left(1 - \frac{2}{n-i+1}\right)$$

$$= \left(\frac{\cancel{n-2}}{n}\right)\left(\frac{\cancel{n-3}}{n-1}\right)\left(\frac{n-4}{\cancel{n-2}}\right) \cdots \left(\frac{2}{4}\right)\left(\frac{1}{3}\right)$$

$$= \frac{2}{n(n-1)}$$

So the probability of failure is $\leq 1 - \frac{2}{n(n-1)}$.

If we repeat the algorithm t times and pick the smallest cut, the probability of failure is at most $\left(1 - \frac{2}{n(n-2)}\right)^t \leq \left(1 - \frac{2}{n^2}\right)^t$

Since $(1+\frac{c}{x})^x \le e^c$, with $t = n^2 \ln n$ the Probability of failure is at most $e^{-2\ln n} = O(n^{-2})$

- Total time for this algorithm: $O(n^4 \ln n)$.
- This is not the most efficient but quite simple.

Deviation Bounds

Example: Coupon Collector's Problem

- A deck of n cards, each time randomly pick one card, note the value, and put back in the same place
- Expected number of rounds before each card is drawn at least once? (think of coupons)
- X : # of rounds before all coupon types collected
- X_i : # of rounds till the next new coupon when i coupons collected.

$$E[X] = \sum_{i=0}^{n-1} E[X_i]$$

- Probability of drawing a new coupon when i

- Probability of drawing a new coupon when i already seen: $P_i = \frac{n-i}{n}$

So $E[X_i] = \frac{n}{n-i} \longrightarrow$

$$E[X] = \sum_{i=0}^{n-1} \frac{n}{n-i} = n \sum_{i=1}^{n} \frac{1}{i} = n H_n = n \ln n + \Theta(n)$$

- Using Markov's: $\Pr[X \geq 2E[X]] = \Pr[X \geq 2n H_n] \leq \frac{1}{2}$

- Using Chebychev's:

easy to show (try doing it!): $Var[X_i] = \frac{1-P_i}{P_i^2} \leq \frac{1}{P_i^2}$

$$Var[X] = \sum_{i=0}^{n-1} Var[X_i] \leq \sum_{i=0}^{n-1} \left(\frac{n}{n-i}\right)^2 = n^2 \sum_{i=1}^{n} \frac{1}{i^2} \leq \frac{n^2 \pi^2}{6}$$

$$\Pr[(X - E[X]) \geq \underbrace{n H_n}_{\lambda}] \leq \frac{\overset{Var}{O(n^2)}}{\underbrace{n^2 \ln^2 n}_{\lambda^2}} = O\left(\frac{1}{\ln^2 n}\right).$$

Chernoff-Hoeffding's Bound

- One of the strongest (and perhaps most useful) tail bounds for "independent" var's.

Chernoff-Hoeffding: Assume $X_1 \cdots X_n$ are independent random variables take values in $[0,1]$. Then for $X = \sum_{i=1}^{n} X_i$, $\mu = E[X]$:

For $X = \sum_{i=1} X_i$, $\mu = E[X]$:

1) for any $0 < \delta$: $\Pr[X \geq (1+\delta)\mu] \leq \left(\frac{e^{\delta}}{(1+\delta)^{(1+\delta)}}\right)^{\mu}$

2) for any $0 < \delta \leq 1$: $\Pr[X \geq (1+\delta)\mu] \leq e^{-\mu\delta^2/3}$

3) for any $r \geq 6\mu$: $\Pr[X \geq r] \leq 2^{-r}$.

Proof: For now suppose X_i Bernouli random variable $X_i \in \{0,1\}$ with $E[X_i] = \mu_i$

For all $t > 0$:

$$\Pr[X \geq (1+\delta)\mu] = \Pr[tX \geq t(1+\delta)\mu]$$
$$= \Pr[e^{tX} \geq e^{t(1+\delta)\mu}].$$

Using Markov's:

$$\Pr[e^{tX} \geq e^{t(1+\delta)\mu}] \leq \frac{E[e^{tX}]}{e^{t(1+\delta)\mu}} = \frac{E[\prod_i e^{tX_i}]}{e^{t(1+\delta)\mu}}$$
$$= \frac{\prod_i E[e^{tX_i}]}{e^{t(1+\delta)\mu}}$$

Note: $e^{tX_i} = \begin{cases} e^t & \mu_i \\ 1 & 1-\mu_i \end{cases}$ and $1+x \leq e^x$

So $E[e^{tX_i}] = e^t\mu_i + (1-\mu_i) = 1 + \mu_i(e^t - 1) \leq e^{\mu_i(e^t-1)}$

$$\Pr[X \geq (1+\delta)\mu] \leq \frac{\prod_i [e^{\mu_i(e^t-1)}]}{\ldots}$$

$$\Pr[X \geq (1+\delta)\mu] \leq \frac{\prod_i [e \qquad]}{e^{t(1+\delta)\mu}}$$

$$= \frac{e^{(e^t-1)\sum_i \mu_i}}{e^{t(1+\delta)\mu}} = \frac{e^{(e^t-1)\mu}}{e^{t(1+\delta)\mu}}$$

- let $t = \ln(1+\delta)$: $\Pr[X \geq (1+\delta)\mu] \leq \left(\frac{e^{\delta}}{(1+\delta)^{(1+\delta)}}\right)^{\mu}$
- Assumption on $X_i \sim \text{Bernoulli} \in \{0,1\}$ can be removed.

Corollary: For $0 < \delta < 1$:

$$\Pr[|X - \mu| \geq \delta\mu] \leq 2e^{-\mu\delta^2/3}.$$

- So with high probability $X \sim (1 \pm \delta)\mu$

Example: Coin flips

- We randomly flip n unbiased coins. Let $X_i = 1$ iff i coin turns heads and $X = \sum X_i$

$\mu = E[X] = \frac{n}{2}$

- Using Markov's: let $\lambda = \mu$, $\Pr[X > \lambda + \mu] \leq \frac{1}{2}$
- Using Chebyshev's: $Var[X] = \frac{n}{4}$, let $\lambda = \sqrt{n}$

Then $\Pr[X > \mu + \lambda] \leq \frac{\frac{n}{4}}{\lambda^2} \leq \frac{1}{4}$

- Using Chernoff bound:

$$\Pr[X \geq \mu + \lambda] = \Pr\left[X \geq \mu\left(1 + \frac{\lambda}{\mu}\right)\right] \leq e^{-\lambda^2/3\mu}$$

$\lambda = \sqrt{3n \ln n}$, $\frac{\lambda^2}{3\mu} = 2 \ln n \longrightarrow$

$$\Pr[X \geq \mu + \sqrt{3n \ln n}\,] \leq e^{-2\ln n} = \frac{1}{n^2}$$