CS121 Lecture 20: Randomized Algorithms Nov. 7

Midterm

- Lectures 11 to 21 (inclusive)
- Focus on time complexity (P, EXP, NP, P/poly, NPC)
- Only true/false on randomized computation (BPP)
- Recommend starting HW5

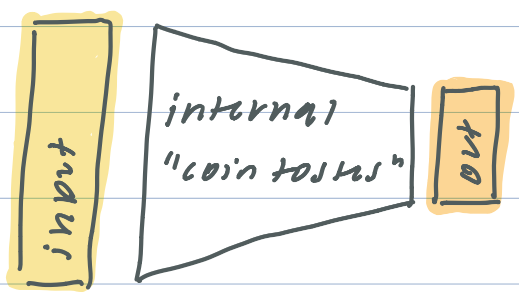
Informal

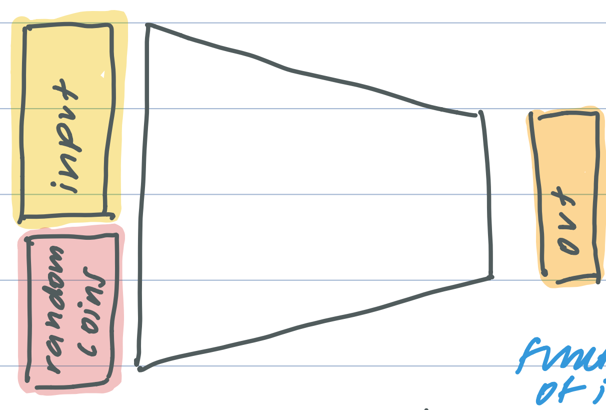
foo = flipping coin $\rightarrow$ foo $\sim \{0,1\}$

through repitition: foo $\sim \{0,1\}^n$ or $[0,1]$

Randomized Algorithms

Equivalent views





- input $x \in \{0,1\}^n$
- alg $A(x)$ has special operation $r_i \leftarrow RAND()$
  $r_i \sim \{0,1\}$

- input $x \in \{0,1\}^n$
- choose $r \sim \{0,1\}^{m}$ (function of input length)
- run deterministic algo $A(x,r)$

output = Alg(input, randomness)

Computing a Function

Randomized Algorithm ALG computes F if for every input X:

arbitrary, but > 0.5

$$\Pr[ALG(X) = F(X)] \geq \frac{2}{3}$$

probability over the randomness of the algorithm, not the input

not random input — has to work in the worst case

Approximation of MAXCUT (amplification of 1-sided error)

Input: $G = (V, E)$

Output: Partition of V maximizing # of crossing edges

Def. $OPT(G) = \max_{S \subseteq V} |E(S, \bar{S})|$ to be max # of cut edges.

If $P \neq NP$, no poly-time computes $OPT(G)$ / produces cut achieving it.

Show: poly-time randomized algorithm that w/ prob ≥ 0.99 outputs cut S that cuts at least $0.5 \cdot OPT(G)$ edges.

Thm. $\exists$ randomized poly-time algorithm A s.t. w/ prob ≥ 0.99

$$A(G) = S \text{ s.t. } |\underline{E(S, \bar{S})}| \geq |E|/2$$

number of edges cut

Lemma: $\exists$ randomized poly-time algorithm A s.t.

if $S = A(G)$ then $\mathbb{E}[|E(S,\bar{S})|] \geq |E|/2$

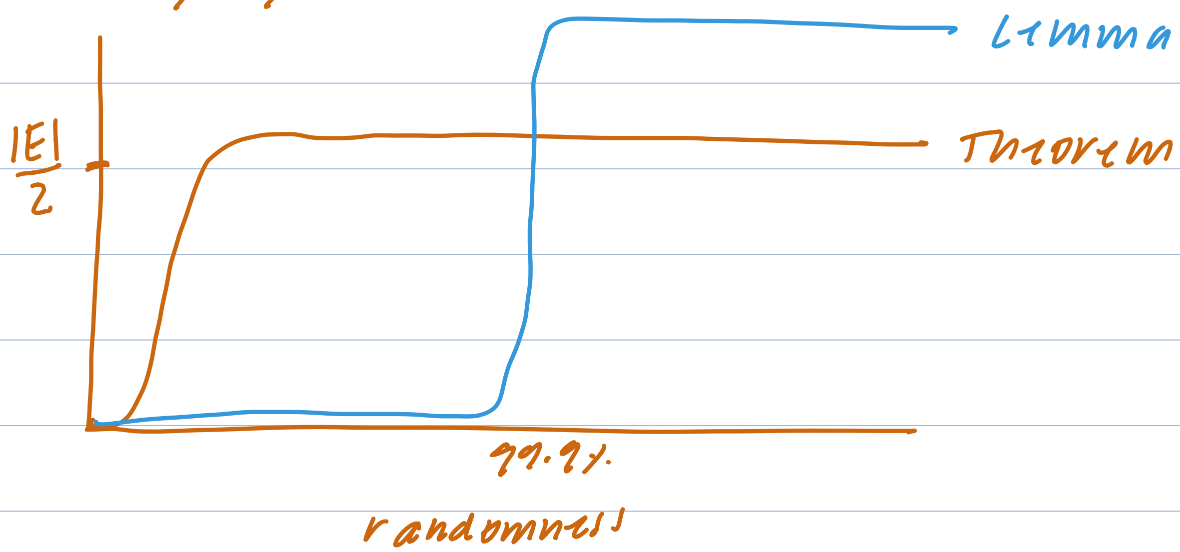
over randomness of A

Q1.

Expectation vs High Probability

some very low, some very high, on average good

always over some threshold of good



Q2. $a, b \sim \{0,1\}$. What is $Pr[a \neq b]$?

$$\left(\frac{2}{2}\right)\left(\frac{1}{2}\right) = \boxed{\frac{1}{2}}$$

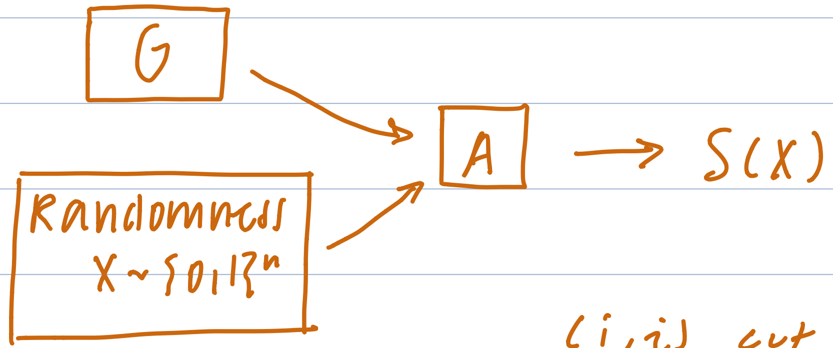
<u>Proof</u>. Given G on n vertices, A picks $X \sim \{0,1\}^n$ and outputs $S = \{i \mid x_i = 1\}$.

For every edge $(i,j) \in E$, define $X_{i,j} = \begin{cases} 1, & X_i \neq X_j \\ 0, & X_i = X_j \end{cases}$

Q3. What is $\mathbb{E}[X_{i,j}]$?

$$1 \cdot Pr(X_i \neq X_j) + 0 \cdot Pr(X_i = X_j) = 1 \times \frac{1}{2} + 0 \times \frac{1}{2} = \boxed{\frac{1}{2}}$$

Q4. Prove that $|E(S, \bar{S})| = \sum_{(i,j) \in E} X_{i,j}$.



(i,j) cut by

$\forall$ edge (i,j) in graph. If $i \neq j$. then edge will be cut, and if $i = j$, then it won't.

$$\mathbb{E}[\#\text{edges cut}] = \sum_{(i,j) \in E} \mathbb{E}[X_{i,j}] = |E| \cdot \tfrac{1}{2} \quad \blacksquare$$

Theorem follows by linearity of expectation.

From expectation to high probability

success amplification

Given: Poly-time alg A s.t. $\mathbb{E}[\text{val}(A(G))] \geq k$

Goal: Poly-time alg B s.t. $\Pr[\text{val}(B(G)) \geq k] \geq 0.99$

Algorithm B

Input: G

\# of edges $\cdot$ 1000

for $i = 1 \ldots 1000m$:

$S_i \leftarrow A(G)$ ← fresh randomness every time

return S_i maximizing edges cut (value)

Lemma. $\Pr[\text{val}(A(G)) \geq k] \geq 1/m$

Q5. Prove that Lemma → $\Pr[\text{val}(A(G)) \geq k] \geq 0.99$

E_1 ~~~~~~ E_{1000m}

where E_i = event that cut in i^{th} iteration had value less than k.

$Pr[\text{"A succeeds"}] \geq 1/m$ by Lemma

$Pr[\text{"A fails"}] \leq 1 - 1/m$

iid

$$Pr[A \text{ fails } 1000m \text{ times}] \leq (1 - 1/m)^{1000m} \approx e^{-1000} < 0.01$$

$$Pr[E_1 \cap E_2 \cap \dots E_{1000m}] = \prod_{i=1}^{1000m} Pr[E_i]$$

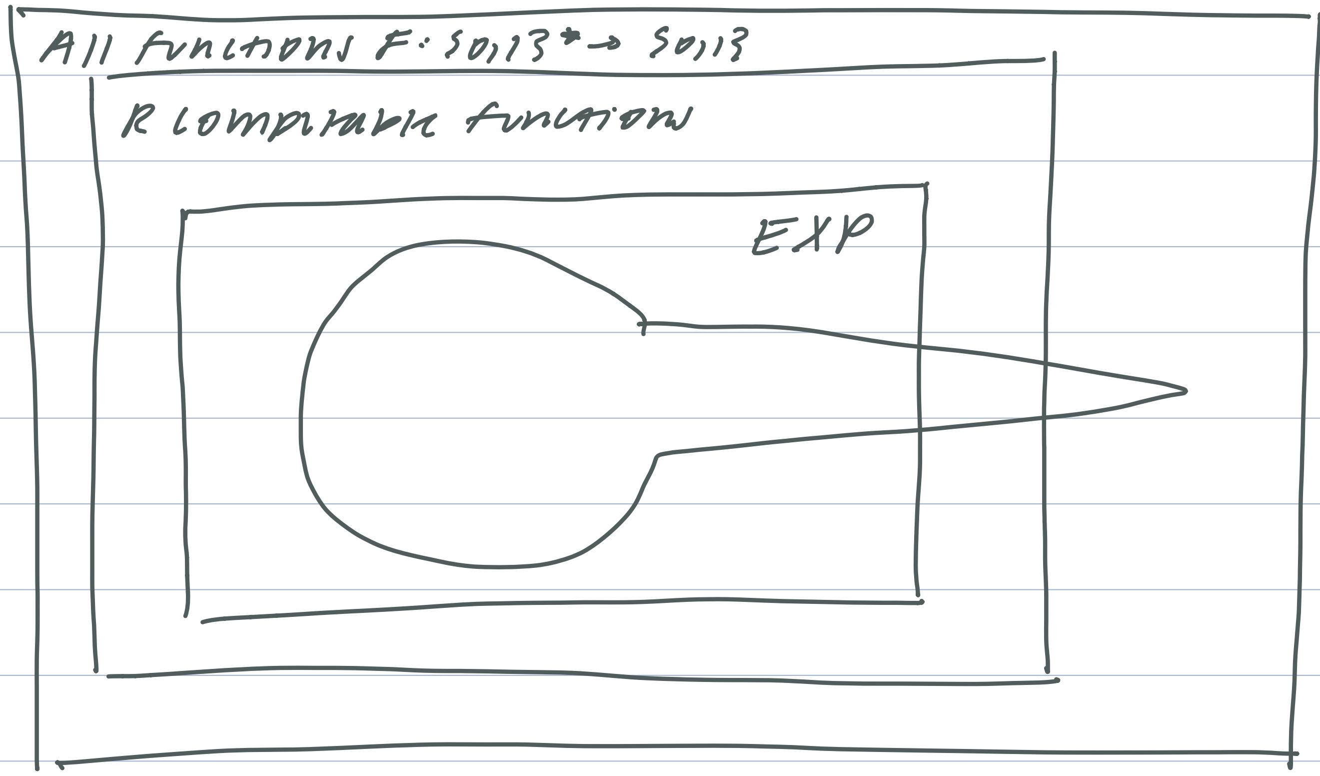
__Pf.__ Suppose $Pr[val(A(b)) \geq k] < \frac{1}{m}$ then

$$\mathbb{E}[val(A(b))] < \underbrace{\frac{1}{m} \cdot m}_{\substack{\text{contribution} \\ \text{from} \\ val(A(b)) \geq k}} + \underbrace{1 \cdot (k-1)}_{\substack{\text{contribution} \\ \text{from} \\ val(A(b)) < k-1}} = k$$

$\blacksquare$

More amplification

If we repeat $t \cdot 1000m$ times, probability of value $< k$ is $<< 2^{-t}$.



All functions $F: \{0,1\}^* \to \{0,1\}$
R computable functions
EXP