Today
- Administrivia
- Introduction
- Power of Randomness (examples)

CSS.413.1
Pseudorandomness
Lecture 01 (2021-08-24)
Instructor: Prahladh Harsha.

Administrivia

Acadly Webpage

Tu-Thurs - 09:30 - 11:00

Grading Policy: 4 problem sets.

Introduction:

Randomness: (1) Is it useful?

- Algorithmic Design
- Cryptography
- Combinatorial Constructions

(2) Does Randomness exist?

- Impure coin.
- Correlated coins

$\left.\begin{array}{l}\text{Impure coin.}\\\text{Correlated coins}\end{array}\right\}$ cope with this?

(3) Do we really need randomness?

- can we eliminate/reduce randomness.

Today: Power of Randomness

Application 1: Equality Protocol

Alice $\qquad\qquad$ Bob.

$x \in \{0,1\}^n$ $\qquad \rightleftarrows \rightleftarrows$ $\qquad y \in \{0,1\}^n$

$x = y$?

Qn: How many bits must Alice & Bob exchange to check if their i/ps are equal?

Easy: $(n+1)$-bit deterministic protocol.

Surprisingly: Tight, ie any det protocol that computes EQ requires $n+1$ bits in the worst case.

Allow Random Coins.

—— r —— (sequence of ind rand coins $r \in \{0,1\}^n$)

Alice $\qquad\qquad$ Bob

$x \in \{0,1\}^n$ $\qquad\qquad$ $y \in \{0,1\}^n$

$a = \langle x, r\rangle = \sum x_i \cdot r_i \cdot \pmod 2$ $\qquad$ $b = \langle y, r \rangle$

$\xrightarrow{a}$

$\xleftarrow{0/1}$ $\qquad$ $a = b$?

Case: $x = y$, $\Pr_r[\text{Protocol is correct}] = 1$

Case: $x \neq y$ $\Pr_r[\text{Protocol is wrong}]$

$$= \Pr_r[\langle x, r\rangle = \langle y, r\rangle]$$

$$= \Pr_r[\langle x-y, r\rangle = 0 \pmod 2]$$

$$\star \quad = \Pr_r[\langle z, r\rangle = 0 \pmod 2]$$

where $z = x - y \neq 0^n$

Since $z \neq 0^n$; $\exists\, i \in [n]$, $z_i = 1$

$$\star \quad = \Pr_r[z_1 r_1 + z_2 r_2 + \ldots + z_i r_i + \ldots + z_n r_n = 0]$$

$$= \Pr_r[r_i = z_1 r_1 + \ldots + z_{i-1} r_{i-1} + z_{i+1} r_{i+1} \ldots + z_n r_n]$$

$$= 1/2.$$

Concl: There is a constant bit randomized protocol for EQ (in the shared random string model)

Application 2: Ramsey Theory

Erdős-Szekeres:
Every n-vertex simple graph has either a clique of size $\frac{1}{2}\log n$ or

an independent set of size $\frac{1}{2}\log n$

(ie, $\alpha(G) \geq \frac{1}{2}\log n$ or $\alpha(\bar{G}) \geq \frac{1}{2}\log n$)

Qn: How tight is the above?

[Erdös] There is an n-vertex simple graph G s.t.

$$\alpha(G) \leq 2\log n \quad \& \quad \alpha(\bar{G}) \leq 2\log n.$$

Proof (via probabilistic method).

$$\Pr_{G \sim \{0,1\}^{\binom{n}{2}}}\left[\alpha(G) \leq 2\log n \;\&\; \alpha(\bar{G}) \leq 2\log n\right] > 0.$$

Open: Come up w/ an alternate "explicit" construction of such a graph.

Recently, Chattopadhyay Zuckerman & Cohen

$\exp((\log\log n)^c)$ for some $c > 1$.

Application 3: Primality Testing

Problem: Given a positive integer n (in binary), check if n is prime or composite?

⋮

Miller: Extended Riemann Hypothesis
$\Downarrow$
det poly algorithm for primality.

Miller-Rabin: Randomized polynomial time alg. MR

n - prime $\qquad \Pr_r [MR(n,r) = \text{prime}] = 1$

n - comp $\qquad \Pr_r [MR(n,r) = \text{prime}] < \frac{1}{100}$

Solovay-Strassen: randomized poly time algorithm.

2002: Agrawal, Kayal & Saxena
- deterministic polytime alg for primality.

Application 4: Generating Primes Efficiently.

Problem: Given a positive integer n (in binary) output a prime number between n & $2n$?

Pick random number in $[n, 2n]$.

$$\Pr[n \text{ is prime}] \approx \frac{1}{\log n}$$

Open: Deterministic procedure to generate large primes.

"pseudo-deterministic" algorithms.

Application 5: Undirected Connectivity (UCONN):

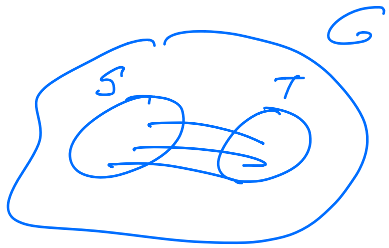
Problem: Given an undirected simple graph $G=(V,E)$ & 2 special vertices s & $t \in V$, are s & t connected?

- Efficient (wrt space) randomized alg for UCONN.
- 2004: Reingold. - deterministic logspace (efficient in space) algorithm for UCONN

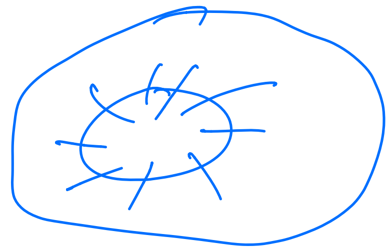
Application 6: MAXCUT

Graph $G=(V,E)$

$S,T \subseteq V$ $\quad cut(S,T) = \{\{u,v\} \mid u \in S \text{ \& } v \in T\}$



$\text{cut}(S) = \text{cut}(S, V \setminus S)$



Problem: Given a simple graph $G = (V, E)$, find a cut S that maximizes $|\text{cut}(S)|$.

NP-hard (Karp's list of NP-complete problems)

Approximation to MAXCUT

<u>½-apprx MAXCUT</u> (using randomness).

Input: $G = (V, E)$

Alg: Output $S \subseteq V$.

$$\mathop{\mathbb{E}}_{S}\left[|\text{cut}(S)|\right] = \mathop{\mathbb{E}}_{S}\left[\sum_{e \in E} \mathbb{1}\left[e \text{ is cut by } (S, V \setminus S)\right]\right]$$

$$= \sum_{e \in E} \mathop{\mathbb{E}}_{S}\left[\mathbb{1}\left[e \in \text{cut}(S, V \setminus S)\right]\right]$$

$$= \sum_{e \in E} \Pr_{S}\left[e \in \text{cut}(S, V \setminus S)\right]$$

$$= \sum_{e \in E} \frac{1}{2} = \frac{|E|}{2}.$$

$\boxtimes$