Randomized Computation

(BPP, RP, ZPP)

Randomize T.M

1.

2 . . . . . Goto either step a or b randomly.

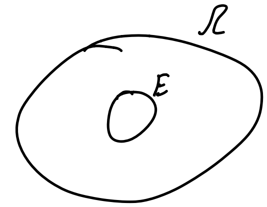
3

x ——— x

Ω

- $\forall \; a \in \Omega \quad 0 \leq Pr[a] \leq 1$
- $\sum_{a \in \Omega} Pr[a] = 1$.

Event

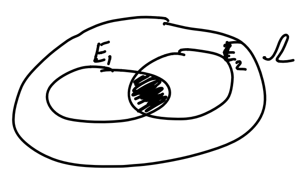


$E \subseteq \Omega$

$Pr[E] = \sum_{a \in E} Pr[a].$

Conditional Probability

$E_1, E_2 \qquad Pr[E_1 | E_2] = \frac{Pr[E_1 \cap E_2]}{Pr[E_2]}$



Random variable

$X : \Omega \to \mathbb{R}$

$\mathbb{E}[X] = \sum_{a} Pr[a] \cdot X(a)$.

Linearity of Expectation

If X, Y are random variables, $\mathbb{E}[X+Y] = \mathbb{E}[X] + \mathbb{E}[Y]$.

$A_1, \dots, A_{200}$

$$X_i = \begin{cases} 1 & \text{if } A_i, A_{i+1}, \dots, A_{i+6} \text{ are all heads} \\ 0 \end{cases}$$

$$\mathbb{E}[X_1 + \dots + X_{194}] = \mathbb{E}[X_1] + \mathbb{E}[X_2] + \dots + \mathbb{E}[X_{194}]$$
$$= \frac{194}{128}.$$

Randomized Algorithms

$A \cdot B$ ($n \times n$ matrices)

- n^3 time algorithm
- $n^{2.34\dots}$
- I tried to prove "no $O(n^2)$ algorithm".

Goal: $A \cdot B \stackrel{?}{=} C$

Algorithm:

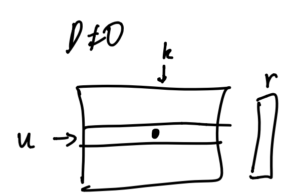
- Sample column vector $r \in \{0,1\}^n$
- Check that $A(Br) = Cr$

Claim: If $AB = C$ then $ABr = Cr$. ✓

Claim: If $AB \neq C$ then $\Pr_r[ABr = Cr] \leq 1/2$.

Pf: Suppose $AB \neq C$.

$$ABr - Cr$$
$$= (AB - C) \cdot r$$
$$= Dr$$



If $u_k \neq 0$ then $\Pr\left[\sum_i u_i r_i = 0\right] \leq 1/2$

$$\sum_i u_i r_i = u_k r_k + \sum_{i \neq k} u_i r_i$$

For every fixed choice of $r_1, \ldots, r_{k-1}, r_{k+1}, \ldots, r_n$

$$\Pr_{r_k}\left[u_k r_k = - \sum_{i \neq k} u_i r_i \right] \leq 1/2 .$$

$$\Pr\left[\sum_i a_i r_i = 0\right] = \sum_{a_1, \ldots, a_{k-1}, a_{k+1}, \ldots, a_n} \Pr\left[\sum_i a_i r_i = 0 \;\middle|\; \begin{matrix} r_1 = a_1 \\ \vdots \\ r_{k-1} = a_{k-1} \\ r_{k+1} = a_{k+1} \\ \vdots \\ r_n = a_n \end{matrix}\right] \cdot \Pr\begin{bmatrix} r_1 = a_1 \\ \vdots \\ r_n = a_n \end{bmatrix}$$

$$\leq \sum_{\ldots} 1/2 \cdot 2^{-(n-1)}$$

$$= 1/2 .$$

$$\frac{a \vee \bar{b}}{b \Rightarrow a}$$

2-SAT

$$\phi = (x_1 \vee \bar{x}_2) \wedge (x_5 \vee \bar{x}_1) \wedge \ldots$$

Algorithm:

0. Start with $x = 0$
1. If $\phi(x) = 1$, we halt.
2. Otherwise find a clause that is not satisfied
3. Randomly flip a variable of that clause.
4. Goto 1.

Claim: If ϕ is satisfiable, expected # of step is $O(n^2)$.

Pf: Suppose $\exists\, y$ $\phi(y) = 1$.

$$|x-y| = \sum_i |x_i - y_i|$$

In each step $|x-y|$ is reduced by 1 with prob $\geq 1/2$.

$t_i = \mathbb{E}[\#\text{ steps to hit } 0 \text{ starting from } i]$.

$$t_0 = 0.$$

$$t_n = t_{n-1} + 1 \Rightarrow t_n - t_{n-1} = 1$$

$$t_i = 1 + \frac{t_{i-1}}{2} + \frac{t_{i+1}}{2}$$

$$\Rightarrow 2t_i = 2 + t_{i-1} + t_{i+1}$$

$$\Rightarrow (t_i - t_{i-1}) = (t_{i+1} - t_i) + 2$$

$$t_n = (t_n - t_{n-1}) + (t_{n-1} - t_{n-2}) + \cdots + (t_1 - t_0)$$

$$= 1 + 3 + 5 + \cdots$$

$$= \sum_{j=1}^{n} (2j-1) = 2 \cdot \sum_{j=1}^{n} j - n$$

$$= 2 \cdot \binom{n}{2} - n$$

$$= n \cdot (n+1) - n$$

$$= n^2.$$