Puzzle:

3 people brought into a room

Hat placed on each person's head: Red or Blue equally likely

Each person sees colors of other people's hats, but not their own

Each person, without communication says: R, B or pass

All 3 shot unless

they can agree on a strategy ahead of time

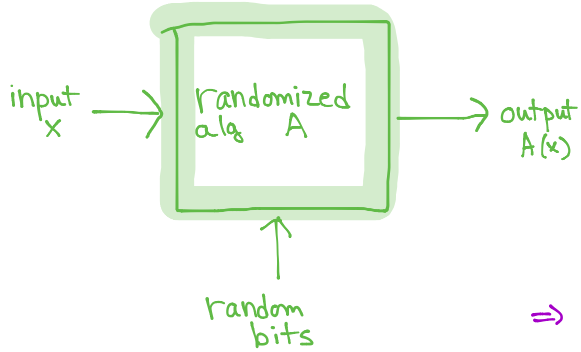
- at least one doesn't pass
- everyone who doesn't pass is right.

Strategy 1: each person guesses $\Rightarrow Pr(\text{not all shot}) = \frac{1}{8}$

Strategy 2: 2 pass, 1 guesses $\Rightarrow Pr(\text{not all shot}) = \frac{1}{2}$

Is there a better strategy?

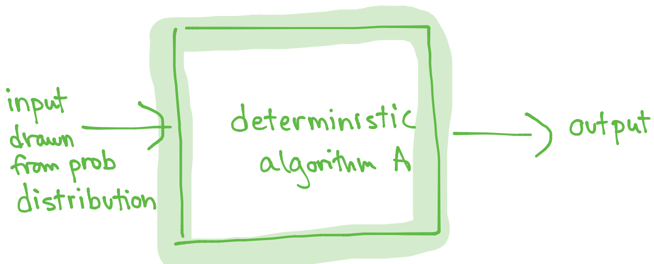
Randomized Algorithms & Probabilistic Analysis of Algorithms....



Model of computation:

standard model (TM, RAM)
with additional input consisting
of stream of perfectly random bits.

⇒ behavior can vary on fixed input
– running time on particular input
is a random variable



Example of difference:

quicksort with randomly
selected pivots vs
QS where input is random π

again, performance of algorithm
is a random variable

also other random structures: random graphs, random boolean formulas, etc.

Why randomized algs?

- often simplest or fastest
- fun!!!

Matrix-Product Verification

[MU] 1.3 [MR] 7.1

Field F:
set with 2 operations
addition, multiplication
has all the properties of $\mathbb{R}$ or rational #'s
e.g. commutativity, associativity
additive/mult. inverses
identity elts for +, ·

$\boxed{\text{Example: } GF(2)}$
addition XOR (addition mod 2)
multiplication AND

Given $n \times n$ matrices A, B, C over field F

Told $AB = C$

Goal: to verify this identity

Obvious method: matrix multiplication

$O(n^{2.376})$

[Freivalds Alg] simple & elegant

one of first published uses of randomization in algs

> Pick random vector $\vec{r} = (r_1, r_2, \dots, r_n) \in \{0,1\}^n$
>
> each r_i indep, equally likely to be 0 or 1
>
> 0: additive identity of field
>
> 1: multiplicative identity of field
>
> compute $A(Br) = z$
>
> If $Cr = z$
>
> then output "yes, $AB = C$"
>
> else output "no"

Running Time:

Errors:

Claim: $\Pr(\text{output an incorrect answer}) \leq \frac{1}{2}$

Proof: Define $D = AB - C$

Suppose $D \not\equiv 0$

Then $\exists$ entry, say (i,j) s.t. $d_{ij} \neq 0$

$$\Pr(Dr = 0) \leq \Pr\left(\sum_k d_{ik} r_k = 0\right)$$

$$= \Pr\left(d_{ij} r_j = -\sum_{k \neq j} d_{ik} r_k\right)$$

$$= \Pr\left(r_j = \frac{-\sum_{k \neq j} d_{ik} r_k}{d_{ij}}\right)$$

Example of simple but powerful principle of deferred decisions

multiple r.v.'s - think of setting some of them first and deferring setting rest until later in analysis

Formally, use law of total probability; condition on values of vars set 1st

$$\Pr\left(r_j = \frac{-\sum_{k\neq j} d_{ik} r_k}{d_{ij}}\right)$$

$$= \sum_{\substack{(x_1,\dots,x_{i-1},x_{i+1},\dots x_n)\\ \in\{0,1\}^{n-1}}} \underbrace{\Pr\left(r_j = \frac{-\sum_{k\neq j} d_{ik} r_k}{d_{ij}} \;\middle|\; \underbrace{\begin{matrix}(r_1,\dots,r_{i-1},r_{i+1},\dots r_n)\\ = (x_1,\dots,x_{i-1},x_{i+1},\dots x_n)\end{matrix}}_{A}\right)}_{\leq \frac{1}{2}} \Pr(A)$$

$$\leq \sum_{\substack{(x_1,\dots,x_{i-1},x_{i+1},\dots x_n)\\ \in\{0,1\}^{n-1}}} \frac{1}{2}\Pr(A) = \frac{1}{2}$$

If want to reduce probability of error, can do so at expense of small ↑ in running time

1. Run alg k times
2. Output yes if get yes all k times

$$\Pr(\text{error}) \leq \frac{1}{2^k}$$

by independence of trials.

Fingerprinting [MR] 7.4 [CG] 2.2.1

A & B each have large DB, separated by long distance
↓ a ↓ b both n-bit strings

want to check if $a = b$?

Deterministically n bits of communication necessary

Next: randomized protocol that uses $O(\log n)$ bits of communication

A picks prime $p \in [2..x]$ u.a.r. → to be determined

A sends $(p, a \bmod p)$ to B

B computes $b \bmod p$

If $a \bmod p = b \bmod p$, B sends back "yes", else "no"

Note: need random prime ...

Always gives right answer if $a=b$.

may give wrong answer if $a \neq b$

Suppose $a \neq b$

$Pr(a \bmod p = b \bmod p) = Pr(\, a-b$ is multiple of $p)$

$$= \frac{\#\text{ distinct primes that divide } a-b}{\#\text{ primes in } [2..x]}$$

$$\leq \frac{\#\text{ distinct primes that divide } a-b}{\#\text{ primes in } [2..x]}$$

$\leq n$

Each prime ≥ 2
can't multiply $> n$
together before get $> 2^n$

<u>Prime # Thm:</u>

$\#\text{ primes} \leq x \approx \frac{x}{\log x}$

$\geq 1.26 \frac{x}{\ln x} \quad \forall x \geq 17$

$$\leq \frac{n \ln x}{1.26\, x}$$

choosing $x = \frac{c}{1.26} n \ln n$

$$\leq \quad \frac{1}{c} \frac{\ln x}{\ln n} = \frac{1}{c} + o(1)$$

bits transmitted

$= 2 \log x = O(\log n)$

Example: $n = 2^{23} \sim 1$ MByte $\qquad x = 2^{32}$ (fingerprints are 32 bit words)

$\Pr(\text{error}) < 0.0035$

Max Cut [MU] 6.2.1 [CG] 1.4.1

simple randomized alg

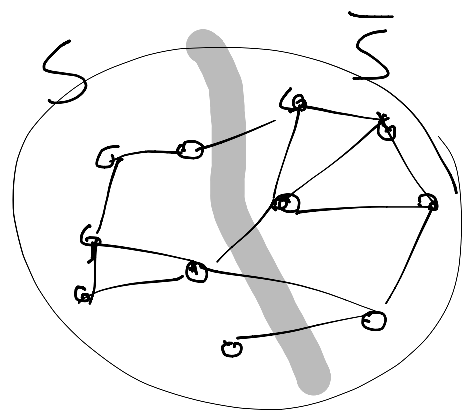
illustration of $\boxed{\text{probabilistic method}}$

use probabilistic argument to prove non-probabilistic mathematical thm

Defn cut in graph: partition of nodes into 2 sets S and $\bar{S}$

An edge crosses cut if it has one endpt in S & one in $\bar{S}$

Thm: In any graph $G=(V,E)$, $\exists$ cut s.t. at least $\frac{1}{2}$ edges cross cut.



Proof technique: show that if we pick a random cut, the exp # of edges that cross cut is $\geq \frac{1}{2}|E|$

Pick cut u.a.r. $\forall v \in V$, flip fair coin $\begin{cases} H \to v \in S \\ T \to v \in \bar{S} \end{cases}$

Let $X_e = \begin{cases} 1 & e \text{ crosses cut} \\ 0 & \text{o.w.} \end{cases}$

$X = \sum_{e \in E} X_e$ # edges crossing cut $E(X) = ?$

$E(X) = E(\sum_{e \in E} X_e) = \sum_{e \in E} E(X_e) = \frac{1}{2}|E|$

$\Rightarrow$ sample space must contain at least one cut

in which $\geq \frac{1}{2}$ edges cross cut. o.w. $E(X) < \frac{1}{2}|E|$

Typical example of prob method:

- Not everybody can be below (or above) average
- Collection of objects $\Pr(\exists$ object with property $P) > 0$

$\Rightarrow \exists$ object in collection with property P