ADVANCED ALGORITHMS

Lecture 14: randomized algorithms
ANNOUNCEMENTS

➤ HW 3 out; due Friday after fall break
➤ Look out for project topics! — announcement in a little while
➤ Groups of 2-3
➤ Give a list of >= 5 topics, top choice first
➤ Send email to utah-algo-ta@googlegroups.com, with subject “Project topic”; one email per group; names and UIDs

(3 weeks from now, 1-on-1 meetings about “proposal”)
(last week of October)
Randomness can help design efficient algorithms — examples “polynomial identity” testing, testing if a number is prime, …
TODAY’S PLAN

➤ Randomness can avoid “unlucky” choices — quick sort, hashing
➤ Reasoning about randomized algorithms
➤ Expectations and linearity
REASONING ABOUT RANDOMIZED ALGORITHMS

- A[1, ..., N] is an array in which N/2 elements promised to be zero. Find one i such that A[i] = 0

- Let i be random index; check if A[i] = 0, if so return it;
- Else repeat

Las Vegas algorithm:
- It is never wrong
- Running time can be large if we are unlucky.

Prob. that algorithm has run time \( t \) is

\[
\leq \frac{1}{2^t}
\]
Running time can be varying — expected running time, ...

Success probability

General tradeoff — boosting

Say we have ALG, that has success prob = \( \frac{1}{2} \) & run time = \( T \).

Then, there is an ALG that has success prob = \( 1 - \delta \) & run time \( T \log(\frac{1}{\delta}) \).

Run time depends on the "coin tosses" so far.

Expected value of any real-valued random variable

runtime is a random variable.
\text{\textit{Alg}}' \\
- Pick \( r = \log \frac{1}{\delta} \).

- Repeat \( r \) times:
  - run algorithm \textit{Alg}
  - if it succeeds, output answer

Failure prob. of \textit{Alg}' \leq \frac{1}{2^r} = \delta.
FLASHBACK — SORTING

➤ Consider following algorithm:
  ➤ Pick A[0] as the “pivot”;
  ➤ Partition A[] into “less than pivot” and “greater than pivot”
  ➤ Sort recursively; return appended sorted lists

\[ T(n) = T(a) + T(b) + O(n) \]

\[ a + b = n - 1 \]

\[ \text{worst case: } a = n - 1 \quad ; \quad b = 0 \]
A[0] AS PIVOT — BAD EXAMPLES?

Qn: when is the running time $\sim \frac{n^2}{2}$?

One example: $B[]$ has size $n-1$; $C[]$ has size $O$

$$T(n) = T(n-1) + O(n) \sim n^2.$$  

$$a, b \geq \frac{m}{3}.$$  

$$|B[], |C[]|$$

If $A[0]$ is “almost in the middle” of the sorted version of $A[]$, then algorithm does well.
RANDOM PIVOT


- Probability of lop-sided partition

What is $\text{Prob that } \min(a, b) < \frac{n}{3}$?

Probability of "succeeding" is $\frac{1}{3}$.

(No lopsided split)
Theorem: Expected running time of QuickSort

\[ T \sim O(n \log n) \]

\[ X_n: \text{running time of algorithm on array of size 'n'} \]

\[ \text{range of } X_n = \{1, \ldots, n^2\} \]

\[ \mathbb{E}[X_n] = \sum_{i=1}^{n^2} i \cdot \Pr[X_n = i] \]
Random variable $X$ takes real values depending on choices of algorithm. 

$\mathbb{E}[X] = \sum_i i \cdot \text{Prob}[X=i]$ 

If coin is heads, $X = 1$. 
If coin is tails, $X = 0$. 

$= 0 \cdot \frac{1}{2} + \frac{1}{2} \cdot 1 = \frac{1}{2}$ 

$= 2 \cdot \frac{1}{2} + \frac{1}{2} \cdot 0 = \frac{3}{2}$ 

$= 3.5$ 

$= \frac{7.1 + 0.9 + 6.1}{9} = 0.79$
"PROBABILISTIC RECURRENCE"

[Writing an recurrence for $\mathbb{E}[X_n]$.]

Define $y_n = \mathbb{E}[X_n]$; $Y_n = \sum_{i=1}^{n} \frac{1}{n} \left\{ y_{i-1} + y_{n-i} + n \right\}$. 

The probability of picking $i = \frac{1}{n}$ is

$$\mathbb{E}[X_n] = \sum_{i=1}^{n} \frac{1}{n} \left\{ \mathbb{E}[X_{i-1}] + \mathbb{E}[X_{n-i}] + O(n) \right\}.$$
\[ y_n = \sum_{i=1}^{n} \frac{1}{n} \left( y_{i-1} + y_{n-i} + n \right) \]

where \( y_0 = 0 \).

Can prove by induction that \( y_n \leq 2n \log n \).

Proof works by plugging in \( y_i \leq 2i \log i \).

\[ y_{i-1} + y_{n-i} + n = (i-1) \log (i-1) + (n-i) \log (n-i) + n \]

\[ \leq \log \left[ (i-1) \log n + (n-i) \log n \right] + n \]

\[ = 2(n-1) \log n + n \]

\[ \leq 2n \log n \]
**EXPECTATION IS SMALL => SMALL RUNNING TIME WHP?**

\[
\begin{align*}
\mathbb{E}[X_n] &= \frac{n \log n}{n^2} \\
\sum_{i=1}^{n^2} i \Pr[X_n = i] &\rightarrow \text{Prob. that } X_n > (1000) n \log n \leq \frac{1}{1000}
\end{align*}
\]
MARKOV’S INEQUALITY

If \( X \) is a non-negative random variable,
then \( \Pr \left[ X > c \cdot \mathbb{E}[X] \right] \leq \frac{1}{c} \).

Proof: Suppose \( \Pr \left[ X > c \cdot \mathbb{E}[X] \right] > \frac{1}{c} \).

\[
\mathbb{E}[X] = \sum_{i \geq 0} i \Pr \left[ X = i \right] = \frac{c \cdot \mathbb{E}[X]}{c} = \sum_{i \geq 1} i \Pr \left[ X = i \right] + \sum_{i \geq 0} i \Pr \left[ X = i \right] \]
Consider

\[\sum_{i \geq c \cdot \mathbb{E}[X]} \mathbb{P}(X = i)\]

\[> \sum_{i > c \cdot \mathbb{E}[X]} (c \cdot \mathbb{E}[X]) \cdot \mathbb{P}(X = i)\]

\[= (c \cdot \mathbb{E}[X]) \cdot \mathbb{P}(X > c \cdot \mathbb{E}[X]) \geq \frac{1}{c}\]

\[> \mathbb{E}[X]\]
EXPECTED SIZE OF BIN
LINEARITY OF EXPECTATION
Question: suppose $n$ “numbered” balls are placed on a line from left to right. What is the expected number of balls that are in the “right position”? 