# 15-210
# Parallel and Sequential Algorithms and Data Structures

## Lecture 15

## Probability and Randomized Algorithms

# Synopsis

- Overview of Discrete Probability
- Finding the two largest elements
- Find the $k^{th}$ smallest element.

# Randomized Algorithms

- Exploit randomness during computation
  - Pivot selection in Quicksort
  - Average case analysis
  - Primality testing
- Question: How many comparisons are needed to find the *second* largest number on a sequence of $n$ numbers?
  - Naive algorithm: $2n - 3$ comparisons
  - Divide and Conquer algorithm: $3n/2$ comparisons
  - Simple randomized algorithm: $n - 1 + 2\log n$ comparisons *on the average*.

# Overview of Discrete Probability

- Probabilistic Experiment: outcome is probabilistic.
- Sample Space ($\Omega$): arbitrary and possibly countably infinite set of possible outcomes.
  - Tossing a coin
  - Throwing a die/pair of dice.
- Primitive Event: Any one of the elements of $\Omega$.
- Event: Any subset of $\Omega$
  - First die is a 5
  - Dice sum to 7
  - Any die is even.

# Probability Function

- Probability Function: $\Omega \to [0, 1]$

$$\sum_{e \in \Omega} \mathbf{Pr}[e] = 1$$

- Probability of an event $A$:

$$\sum_{e \in A} \mathbf{Pr}[e]$$

  - Probability of “first die is 4”?
  - Probability of “dice sum to to 4”?

# Random Variables

- Random Variable: $X : \Omega \to \Re$
  - $X$ is the sum of the two die rolls
- Indicator Random Variable: $Y : \Omega \to \{0, 1\}$
  - $Y$ is 1 if the dice are the same, 0 otherwise
  - $Y$ is 1 if the total is larger than 7, 0 otherwise
- For $a \in \Re$, the event "$X = a$" is the set

$$\{\omega \in \Omega \mid X(\omega) = a\}$$

# Expectation

- The expectation of a random variable

$$\mathop{\mathbf{E}}_{\Omega,\mathbf{Pr}[]}[X] = \sum_{e \in \Omega} X(e) \cdot \mathbf{Pr}[e] \,.$$

- The expectation of an *indicator* random variable:

$$\mathbf{E}[Y] = \sum_{e \in \Omega, p(e) = \mathtt{true}} \mathbf{Pr}[e] = \sum_{e \in \Omega} \mathbf{Pr}[\{e \in \Omega \mid p(e)\}] \,.$$

  - $p : \Omega \to \mathtt{bool}$

# INDEPENDENCE

- Events $A$ and $B$ are independent if the occurence of one does not affect the probability of the other

$$\mathbf{Pr}[A \cap B] = \mathbf{Pr}[A] \cdot \mathbf{Pr}[B]$$

  - $A = \{(d_1, d_2) \in \Omega \mid d_1 = 1\}$ and $B = \{(d_1, d_2) \in \Omega \mid d_2 = 1\}$ are independent.
  - $C = \{(d_1, d_2) \in \Omega \mid d_1 + d_2 = 4\}$ is NOT independent of $A$ (Why?)

# Independence

- Events $A_1, \ldots, A_k$ are *mutually independent* if and only if for any non-empty subset $I \subseteq \{1, \ldots, k\}$,

$$\mathbf{Pr}[\bigcap_{i \in I} A_i] = \prod_{i \in I} \mathbf{Pr}[A_i].$$

- Random variable $X$ and $Y$ are independent if fixing one does NOT affect the probability distribution of the other.
  - $X =$ "value of the first die" is independent of $Y =$ "value of the second die".
  - $X$ is NOT independent of $Z =$ "sum of the dice"

# Linearity of Expectations

- Important Theorem: given two random variables $X$ and $Y$

$$\mathbf{E}[X] + \mathbf{E}[Y] = \mathbf{E}[X + Y]$$

- Easy to show!

$$\sum_{e \in \Omega} \mathbf{Pr}[e] X(e) + \sum_{e \in \Omega} \mathbf{Pr}[e] Y(e) = \sum_{e \in \Omega} \mathbf{Pr}[e](X(e) + Y(e))$$

- Expected sum of two dice
  - Consider 36 outcomes and take average
  - Sum expectations for each dice $(3.5 + 3.5 = 7)$

# Linearity of Expectations

- In general, for a binary function $f$ the equality

$$f(\mathbf{E}[X], \mathbf{E}[Y]) = \mathbf{E}[f(X, Y)]$$

  is **not** true in general.
  - $\max(\mathbf{E}[X], \mathbf{E}[Y]) \neq \mathbf{E}[\max(X, Y)]$
  - What is $\mathbf{E}[\max(X, Y)]$?
- $\mathbf{E}[X] \times \mathbf{E}[Y] = \mathbf{E}[X \times Y]$ is true if $X$ and $Y$ are independent.

# Examples

- Toss $n$ coins with probability of heads, $p$. What is the expected value of $X$, the number of heads?

$$
\begin{aligned}
\mathbf{E}[X] &= \sum_{k=0}^{n} k \cdot \mathbf{Pr}[X = k] \\
&= \sum_{k=1}^{n} k \cdot p^k (1-p)^{n-k} \binom{n}{k} \text{(Why?)} \\
&= \sum_{k=1}^{n} k \cdot \frac{n}{k} \binom{n-1}{k-1} p^k (1-p)^{n-k} \qquad \left[\text{because } \binom{n}{k} = \frac{n}{k}\binom{n-1}{k-1}\right] \\
&= n \sum_{k=1}^{n} \binom{n-1}{k-1} p^k (1-p)^{n-k}
\end{aligned}
$$

# EXAMPLES

- Toss $n$ coins with probability of heads, $p$. What is the expected value of $X$, the number of heads?

$$\mathbf{E}[X] = \sum_{k=0}^{n} k \cdot \mathbf{Pr}[X = k]$$

$$\ldots$$

$$= n \sum_{j=0}^{n-1} \binom{n-1}{j} p^{j+1} (1-p)^{n-(j+1)} \qquad [\text{ because } k = j+1 \text{ ]}$$

$$= n \cdot p \sum_{j=0}^{n-1} \binom{n-1}{j} p^{j} (1-p)^{(n-1)-j}$$

$$= n \cdot p \cdot (p + (1-p))^{n-1} \qquad [\text{ Binomial Theorem ]}$$

$$= n \cdot p$$

# Examples

- Toss $n$ coins with probability of heads, $p$. What is the expected value of $X$, the number of heads?
- Using linearity of expectations.
  - $X_i = \mathbb{I}\{i\text{-th coin turns up heads}\}$
  - $X = \sum_{i=1}^{n} X_i$

$$\mathbf{E}[X] = \mathbf{E}\left[\sum_{i=1}^{n} X_i\right] = \sum_{i=1}^{n} \mathbf{E}[X_i] = \sum_{i=1}^{n} p = n \cdot p$$

  - because $\mathbf{E}[X_i] = p$.

# Examples

- A coin has a probability $p$ of coming up heads. What is the expected value of $Y$ representing the number of flips until we see a head?
- Write a recurrence!
  - With probability $p$, we'll get a head and we are done,
  - With probability $1 - p$, we'll get a tail and we'll go back to square one

$$\begin{aligned} \mathbf{E}[Y] &= p \cdot 1 + (1-p)\Big(1 + \mathbf{E}[Y]\Big) \\ &= 1 + (1-p)\,\mathbf{E}[Y] \implies \mathbf{E}[Y] = 1/p. \end{aligned}$$

# Finding the Top Two Elements

```
fun max2(S) = let
   fun replace((m1, m2), v) =
      if v ≤ m2 then (m1, m2)
      else if v ≤ m1 then (m1, v)
      else (v, m1)

    start = if S1 ≥ S2 then (S1, S2) else (S2, S1)

in iter replace start S⟨3, ..., n⟩
end
```

- We will do exact analysis.
- $1 + 2(n - 2) = 2n - 3$ comparisons in the worst case. (Why?)
- A Divide and Conquer algorithm gives $3n/2 - 2$

# WORST CASE ANALYSIS

```
fun max2(S) = let
  fun replace((m1, m2), v) =
    if v ≤ m2 then (m1, m2)
    else if v ≤ m1 then (m1, v)
    else (v, m1)
  start = if S1 ≥ S2 then (S1, S2) else (S2, S1)
in iter replace start S ⟨3, ..., n⟩
end
```

- An already sorted sequence (e.g., $\langle 1, 2, 3, \ldots, n\rangle$) will need exactly $2n - 3$ comparisons.
- But this happens with $1/n!$ chance!

# A Randomized Algorithm

- The worst-case analysis is overly pessimistic.
- Consider the following variant:
- On input of a sequence $S$ of $n$ elements:
  1. Let $T = \texttt{permute}(S, \pi)$, where $\pi$ is a random permutation (i.e., we choose one of the $n!$ permutations).
  2. Run the naïve algorithm on $T$.
- No need to really generate the permutation!
  - Just pick an unprocessed element randomly until all elements are processed.
  - It is convenient to model this by one initial permutation!

# Analysis

```
1  fun max2(S) = let
2    fun replace((m1, m2), v) =
3      if v ≤ m2 then (m1, m2)
4      else if v ≤ m1 then (m1, v)
5      else (v, m1)

6    start = if S1 ≥ S2 then (S1, S2) else (S2, S1)

7  in iter replace start S⟨3, ..., n⟩
8  end
```

- $X_i = 1$ if $T_i$ is compared in Line 4, 0 otherwise.
- $Y$ is the number of comparisons

$$Y = \underbrace{1}_{\text{Line 6}} + \underbrace{n-2}_{\text{Line 3}} + \underbrace{\sum_{i=3}^{n} X_i}_{\text{Line 4}};$$

# Analysis

- This expression in true regardless of the random choice we're making.
- We're interested in computing the expected value of $Y$.
- By linearity of expectation,

$$\begin{aligned}\mathbf{E}[Y] &= \mathbf{E}\left[1 + (n-2) + \sum_{i=3}^{n} X_i\right] \\ &= 1 + (n-2) + \sum_{i=3}^{n} \mathbf{E}[X_i].\end{aligned}$$

# Analysis

- Problem boils down to computing $\mathbf{E}[X_i]$, for $i = 3, \ldots, n$!
- What is the probability that $T_i > m_2$?
  - $T_i > m_2$ holds when $T_i$ is either the largest or the second largest in $\{T_1, \ldots, T_i\}$
- So, what is the probability that $T_i$ is one of the two largest elements in a randomly permuted sequence of length $i$?
  - $\frac{1}{i} + \frac{1}{i} = \frac{2}{i}$
- $\mathbf{E}[X_i] = 1 \cdot \frac{2}{i} = 2/i$

# Analysis

$$
\begin{aligned}
\mathbf{E}[Y] &= 1 + (n-2) + \sum_{i=3}^{n} \mathbf{E}[X_i] \\
&= 1 + (n-2) + \sum_{i=3}^{n} \frac{2}{i} \\
&= 1 + (n-2) + 2\left(\tfrac{1}{3} + \tfrac{1}{4} + \ldots \tfrac{1}{n}\right) \\
&= n - 4 + 2\left(1 + \tfrac{1}{2} + \tfrac{1}{3} + \tfrac{1}{4} + \ldots \tfrac{1}{n}\right) \\
&= n - 4 + 2H_n
\end{aligned}
$$

- $H_n$ is the $n^{th}$ Harmonic number
- $H_n \leq 1 + \log_2 n$
- $\mathbf{E}[Y] \leq n - 2 + 2\log_2 n$

# Finding the $k^{th}$ Smallest Element

- **Input:** a sequence of $n$ numbers (not necessarily sorted)
- **Output:** the $k^{th}$ smallest value in $S$ (i.e., `(nth (sort S) k)`).
- *Requirement:* $O(n)$ expected work and $O(\log^2 n)$ span.
- $k$ is 0-based. (For the third smallest element we set $k = 2$).

- We can't really sort the sequence!

# FINDING THE $k^{th}$ SMALLEST ELEMENT

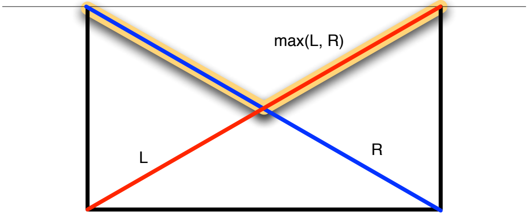
```
fun kthSmallest(k, S) = let
      p = a value from S picked uniformly at random
      L = ⟨ x ∈ S | x < p ⟩
      R = ⟨ x ∈ S | x > p ⟩
   in if (k < |L|) then kthSmallest(k, L)
      else if (k < |S| − |R|) then p
      else kthSmallest(k − (|S| − |R|), R)
```

- Let $X_n = \max\{|L|, |R|\}$

$$W(n) = W(X_n) + O(n)$$
$$S(n) = S(X_n) + O(\log n)$$

# Finding the $k^{th}$ Smallest Element

- We want to find $\mathbf{E}[X_n]$?




$$\mathbf{E}[X_n] = \sum_{i=1}^{n-1} \max\{i, n-i\} \cdot \tfrac{1}{n} \leq \sum_{j=n/2}^{n-1} \frac{2}{n} \cdot j \leq \frac{3n}{4}$$

# Finding the $k^{th}$ Smallest Element

- $\mathbf{E}[X_n] \leq \frac{3n}{4} \Rightarrow$ geometrically decreasing sum $\Rightarrow O(n)$ work.
- What is $\mathbf{Pr}[X_n \leq \frac{3}{4}n]$?
- Since $|R| < n - |L|$,

$$X_n \leq \frac{3}{4}n \Leftrightarrow n/4 < |L| \leq 3n/4$$

and the probability is

$$\frac{3n/4 - n/4}{n} = \frac{n/2}{n} = \frac{1}{2}$$

# Finding the $k^{th}$ Smallest Element

$$\overline{W}(n) = \sum_i \mathbf{Pr}[X_n = i] \cdot \overline{W}(i) + c \cdot n$$

Using stepwise approximation

$$\begin{aligned}
&\leq \mathbf{Pr}[X_n \leq \tfrac{3n}{4}]\overline{W}(3n/4) + \mathbf{Pr}[X_n > \tfrac{3n}{4}]\overline{W}(n) + c \cdot n \\
&= \tfrac{1}{2}\overline{W}(3n/4) + \tfrac{1}{2}\overline{W}(n) + c \cdot n \\
&\implies (1 - \tfrac{1}{2})\overline{W}(n) = \tfrac{1}{2}\overline{W}(3n/4) + c \cdot n \\
&\implies \overline{W}(n) \leq \overline{W}(3n/4) + 2c \cdot n
\end{aligned}$$

- Root Dominated hence solves to $O(n)$.

# Finding the $k^{th}$ Smallest Element

$$S(n) = S(X_n) + O(\log n)$$

$$\begin{aligned}
\overline{S}(n) &=\leq \sum_i \mathbf{Pr}[X_n = i] \cdot \overline{S}(i) + c \log n \\
&\leq \mathbf{Pr}[X_n \leq \tfrac{3n}{4}]\overline{S}(3n/4) + \mathbf{Pr}[X_n > \tfrac{3n}{4}]\overline{S}(n) + c \cdot \log n \\
&\leq \tfrac{1}{2}\overline{S}(3n/4) + \tfrac{1}{2}\overline{S}(n) + c \cdot \log n \\
&\implies (1 - \tfrac{1}{2})\overline{S}(n) \leq \tfrac{1}{2}\overline{S}(3n/4) + c \log n \\
&\implies \overline{S}(n) \leq \overline{S}(3n/4) + 2c \log n
\end{aligned}$$

- This solves to $O(\log^2 n)$.