# A brief introduction to Randomized Algorithms

Ragesh Jaiswal, CSE, UCSD

# Recommended Books

- *Randomized Algorithms* by Rajeev Motwani and Prabhakar Raghavan.
- *Probability and Computing* by Michael Mitzenmacher and Eli Upfal.

## Introduction

- Randomized Algorithms: Algorithms that have additional random input bits.
- Why study randomized algorithms?
  - Simplifies deterministic algorithms.
  - Efficient randomized algorithm for certain problems for which no deterministic algorithms are known.
  - May be used to break symmetry in distributed settings.

# Introduction

- Randomized Algorithms: Algorithms that have additional random input bits.
- Why study randomized algorithms?
  - Simplifies deterministic algorithms.
  - Efficient randomized algorithm for certain problems for which no deterministic algorithms are known.
  - May be used to break symmetry in distributed settings.
- What you should expect to learn in this brief introduction?
  - Basic techniques for using randomness to design algorithms for problems.
  - Techniques for analyzing randomized algorithms.
  - Hash functions, Karger's algorithm, Lovasz Local Lemma(LLL) etc.

- Hashing: A set of $S$ keys from a large universe $U = \{0, ..., m-1\}$ is stored in a table $T = \{0, ..., n-1\}$ using a hash function $h : U \rightarrow T$ so as to minimize the number of collisions. Collisions are resolved using external data structures.
- If $m > n$, then any deterministic function $h$ is bad.

- Hashing: A set of $S$ keys from a large universe $U = \{0, ..., m-1\}$ is stored in a table $T = \{0, ..., n-1\}$ using a hash function $h : U \rightarrow T$ so as to minimize the number of collisions. Collisions are resolved using external data structures.
- If $m > n$, then any deterministic function $h$ is bad.
- Main idea: Choose $h$ randomly from a hash function family $H$.
- Let $H$ consists of all functions from $U$ to $\{0, ..., n-1\}$.
- Consider $t$ insert operations. What is the expected cost of each operation?

- Hashing: A set of $S$ keys from a large universe $U = \{0, ..., m-1\}$ is stored in a table $T = \{0, ..., n-1\}$ using a hash function $h : U \rightarrow T$ so as to minimize the number of collisions. Collisions are resolved using external data structures.
- If $m > n$, then any deterministic function $h$ is bad.
- Main idea: Choose $h$ randomly from a hash function family $H$.
- Let $H$ consists of all functions from $U$ to $\{0, ..., n-1\}$.
- Consider $t$ insert operations. What is the expected cost of each operation? $(1 + t/n)$

# Linearity of Expectation

**Lemma (Linearity of Expectation)**

*For any random variables* $X_1, X_2$ *and constants* $c_1, c_2$, *we have*

$$\mathbf{E}[c_1 X_1 + c_2 X_2] = c_1 \mathbf{E}[X_1] + c_2 \mathbf{E}[X_2]$$

# Linearity of Expectation

**Lemma (Linearity of Expectation)**

*For any random variables $X_1, X_2$ and constants $c_1, c_2$, we have*

$$\mathbf{E}[c_1 X_1 + c_2 X_2] = c_1 \mathbf{E}[X_1] + c_2 \mathbf{E}[X_2]$$

**Hat check problem**

$n$ men go to a party and their hats get mixed up. They randomly pick up a hat. What is the expected number of men who get their own hats?

# Linearity of Expectation

**Lemma (Linearity of Expectation)**

*For any random variables* $X_1, X_2$ *and constants* $c_1, c_2$, *we have*

$$\mathbf{E}[c_1 X_1 + c_2 X_2] = c_1 \mathbf{E}[X_1] + c_2 \mathbf{E}[X_2]$$

**Hat check problem**

$n$ men go to a party and their hats get mixed up. They randomly pick up a hat. What is the expected number of men who get their own hats?

### Coupon Collector Problem

There are $n$ different type of coupons. You get a random coupon every day. What is the expected number of days in which you should collect at least one coupon of each type?

# Linearity of Expectation

## Coupon Collector Problem

There are $n$ different type of coupons. You get a random coupon every day. What is the expected number of days in which you should collect at least one coupon of each type?

- Define $i^{th}$ epoch to be the sequence of days starting the day after the $(i-1)^{th}$ new coupon was collected and ending on the day the $i^{th}$ coupon was collected.
- Define $X_i$ to be a random variable denoting the number of days in the $i^{th}$ epoch. Note that $X_1 = 1$.
- We are interested in knowing the expected value of $X = X_1 + ... + X_n$.
- What is the value of $\mathbf{E}[X_i]$?

# Linearity of Expectation

## Coupon Collector Problem

There are $n$ different type of coupons. You get a random coupon every day. What is the expected number of days in which you should collect at least one coupon of each type?

- Define $i^{th}$ epoch to be the sequence of days starting the day after the $(i-1)^{th}$ new coupon was collected and ending on the day the $i^{th}$ coupon was collected.
- Define $X_i$ to be a random variable denoting the number of days in the $i^{th}$ epoch. Note that $X_1 = 1$.
- We are interested in knowing the expected value of $X = X_1 + ... + X_n$.
- What is the value of $\mathbf{E}[X_i]$? $\mathbf{E}[X_i] = \frac{n}{n-i+1}$
- So, we have:

$$\begin{aligned} \mathbf{E}[X] = \mathbf{E}[X_1 + ... + X_n] &= \mathbf{E}[X_1] + ... + \mathbf{E}[X_n] \\ &= n \cdot (1 + 1/2 + 1/3 + ... + 1/n) \\ &= n \cdot H_n = O(n \cdot \log n) \end{aligned}$$

# Deviation from Expectation

**Theorem (Markov's Inequality)**

*Let $X$ be a non-negative random variable and $a > 0$, then* $\mathbf{Pr}[X \geq a] \leq \frac{\mathbf{E}[X]}{a}$.

**Corollary**

*Let $X$ be a non-negative random variable and $c \geq 1$, then* $\mathbf{Pr}[X \geq c \cdot \mathbf{E}[X]] \leq \frac{1}{c}$.

# Deviation from Expectation

**Theorem (Markov's Inequality)**

*Let $X$ be a non-negative random variable and $a > 0$, then* $\mathbf{Pr}[X \geq a] \leq \frac{\mathbf{E}[X]}{a}$.

**Corollary**

*Let $X$ be a non-negative random variable and $c \geq 1$, then* $\mathbf{Pr}[X \geq c \cdot \mathbf{E}[X]] \leq \frac{1}{c}$.

- Hat-check Problem: What is the probability that at least 10 people out of $n$ get their own hats?
  - $\mathbf{E}[X] = 1$. So, from Markov, we get that $\mathbf{Pr}[X \geq 10] \leq 0.1$.
- Note that
  - $\mathbf{Pr}[\text{everyone gets their own hats}] = \frac{1}{n!}$
  - On the other hand from Markov, we get that $\mathbf{Pr}[X \geq n] \leq 1/n$.

# Deviation from Expectation

**Theorem (Markov's Inequality)**

*Let $X$ be a non-negative random variable and $a > 0$, then* $\mathbf{Pr}[X \geq a] \leq \frac{\mathbf{E}[X]}{a}$.

**Theorem (Chebychev's Inequality)**

*Let $X$ be a random variable and $a > 0$, then* $\mathbf{Pr}[|X - \mathbf{E}[X]| \geq a] \leq \frac{Var[X]}{a^2}$.

# Deviation from Expectation

**Theorem (Chernoff bounds 1)**

*Let $X_1, ..., X_n$ be* independent *0/1 random variables. Let $X = X_1 + ... + X_n$ and $\mu = \mathbf{E}[X]$. Let $\delta > 0$ be any real number. Then $\mathbf{Pr}[X > (1+\delta)\cdot\mu] \leq e^{-f(\delta)\cdot\mu}$, where $f(\delta) = (1+\delta)\ln(1+\delta) - \delta$.*

- Claim 1: $\forall \delta > 0,\ f(\delta) \geq \frac{\delta^2}{2+\delta}$.

**Theorem (Chernoff bound 2)**

*Let $X_1, ..., X_n$ be* independent *0/1 random variables. Let $X = X_1 + ... + X_n$ and $\mu = \mathbf{E}[X]$. Let $\delta > 0$ be any real number. Then $\mathbf{Pr}[X < (1-\delta)\cdot\mu] \leq e^{-g(\delta)\cdot\mu}$, where $g(\delta) = (1-\delta)\ln(1-\delta) + \delta$.*

- Claim 2: $\forall \delta > 0,\ g(\delta) \geq \frac{\delta^2}{2}$.

# Deviation from Expectation

**Theorem (Chernoff bounds special case)**

*Let* $X_1, ..., X_n$ *be* independent $\{\pm 1\}$ *random variables such that for all* $i$, $\mathbf{Pr}[X_i = +1] = \mathbf{Pr}[X_i = -1] = 1/2$. *Let* $X = X_1 + ... + X_n$ *and* $\mu = \mathbf{E}[X]$. *Let* $A > 0$ *be any real number. Then*

$$\mathbf{Pr}[X \geq A] \leq e^{-\frac{A^2}{2n}}.$$

# Birthday Problem

## Birthday Problem

You uniformly sample $q$ items with replacement from a collection of $n$ items. What is the probability that two items are the same?

## Birthday Problem (popular version)

There are $q$ people in a room. What is the value of $q$ such that the probability of two people having the same birthday is at least $1/2$. Each person's birthday is assumed to be a random day in the year.

# Birthday Problem

**Birthday Problem**

You uniformly sample $q$ items with replacement from a collection of $n$ items. What is the probability that two items are the same?

- Let $X_{ij}$ be an indicator random variable that is 1 if the $i^{th}$ and $j^{th}$ person has the same birthday and 0 otherwise.
- Claim 1: $\forall i < j, \mathbf{E}[X_{ij}] = 1/n$.
- Let $X$ denotes the number of distinct pairs of people that have the same birthday.
- Claim 2: $X = \sum_{i<j} X_{ij}$.
- Claim 3: $\mathbf{E}[X] = \frac{q(q-1)}{2} \cdot \frac{1}{n}$ (by linearity of expectation).
- So, if $q \approx \sqrt{2n}$, then $\mathbf{E}[X] > 1$.

## Birthday Problem

You uniformly sample $q$ items with replacement from a collection of $n$ items. What is the probability that two items are the same?

- Let $X_{ij}$ be an indicator random variable that is 1 if the $i^{th}$ and $j^{th}$ person has the same birthday and 0 otherwise.
- Claim 1: $\forall i < j, \mathbf{E}[X_{ij}] = 1/n$.
- Let $X$ denotes the number of distinct pairs of people that have the same birthday.
- Claim 2: $X = \sum_{i<j} X_{ij}$.
- Claim 3: $\mathbf{E}[X] = \frac{q(q-1)}{2} \cdot \frac{1}{n}$ (by linearity of expectation).
- So, if $q \approx c \cdot \sqrt{2n}$, then $\mathbf{E}[X] = 10$.
- Claim 4: $\mathbf{Var}[X_{ij}] = \frac{(n-1)}{n^2}$.
- Claim 5: $\mathbf{Var}[X] = \sum_{i<j} \mathbf{Var}[X_{ij}]$.
- So, $\mathbf{Var}[X] = \frac{q(q-1)(n-1)}{2n^2} = 10 \cdot (1 - 1/n)$ for $q \approx c \cdot \sqrt{2n}$.
- By Chebychev, we get $\mathbf{Pr}[X < 1] \leq \mathbf{Pr}[|X - \mathbf{E}[X]| \geq 9] \leq \frac{10}{81} < \frac{1}{4}$.

# Randomized Quick Sort

# Randomized Quick Sort

## Problem

Sort a given an array of integers containing $n$ distinct integers.

## Algorithm

Randomized-Quick-Sort($A$)

- If $(|A| = 1)$return($A$)
- Randomly pick an index $i$ in the array $A$
- Let $A_L$ denote the array of elements that are smaller than $A[i]$
- Let $A_R$ denote the array of elements that are larger than $A[i]$
- $B_L \leftarrow$ Randomized-Quick-Sort($A_L$)
- $B_R \leftarrow$ Randomized-Quick-Sort($A_R$)
- return$(B_L|A[i]|B_R)$

# Randomized Quick Sort

## Algorithm

`Randomized-Quick-Sort`$(A)$

- If $(|A| = 1)$return$(A)$
- Randomly pick an index $i$ in the array $A$
- Let $A_L$ denote the array of elements that are smaller than $A[i]$
- Let $A_R$ denote the array of elements that are larger than $A[i]$
- $B_L \leftarrow$ `Randomized-Quick-Sort`$(A_L)$
- $B_R \leftarrow$ `Randomized-Quick-Sort`$(A_R)$
- return$(B_L|A[i]|B_R)$

- Let $T(n)$ denote the expected number of comparisons performed.
- Claim 1: $T(n) = (n-1) + \frac{1}{n} \cdot \sum_{i=1}^{n-1}(T(i) + T(n-i-1))$ and $T(1) = 0$.

# Randomized Quick Sort

## Algorithm

`Randomized-Quick-Sort`$(A)$

- If $(|A| = 1)$return$(A)$
- Randomly pick an index $i$ in the array $A$
- Let $A_L$ denote the array of elements that are smaller than $A[i]$
- Let $A_R$ denote the array of elements that are larger than $A[i]$
- $B_L \leftarrow$ `Randomized-Quick-Sort`$(A_L)$
- $B_R \leftarrow$ `Randomized-Quick-Sort`$(A_R)$
- return$(B_L|A[i]|B_R)$

- Let $T(n)$ denote the expected number of comparisons performed.
- Claim 1: $T(n) = (n-1) + \frac{1}{n} \cdot \sum_{i=1}^{n-1}(T(i) + T(n-i-1))$ and $T(1) = 0$.
- So, $T(n) = (n-1) + \frac{2}{n} \cdot \sum_{i=0}^{n-1} T(i)$.
- How do we solve such recurrence relations?

# Randomized Quick Sort

**Algorithm**

`Randomized-Quick-Sort`$(A)$
- If $(|A| = 1)$return$(A)$
- Randomly pick an index $i$ in the array $A$
- Let $A_L$ denote the array of elements that are smaller than $A[i]$
- Let $A_R$ denote the array of elements that are larger than $A[i]$
- $B_L \leftarrow$ `Randomized-Quick-Sort`$(A_L)$
- $B_R \leftarrow$ `Randomized-Quick-Sort`$(A_R)$
- return$(B_L|A[i]|B_R)$

- Here is another way to analyze the algorithm.
- For $i < j$, let $X_{ij}$ be a r.v. that is 1 if a comparison between $A[i]$ and $A[j]$ is made and 0 otherwise.
- Claim 1: $\mathbf{E}[X_{ij}] = \frac{2}{j-i+1}$.
- So, the expected time is:

$$\mathbf{E}\left[\sum_{i<j} X_{ij}\right] = \sum_{i<j} \mathbf{E}[X_{ij}] = \sum_{i=1}^{n} 2 \cdot \left(\frac{1}{2} + \frac{1}{3} + ... + \frac{1}{n-i+1}\right) < 2n \ln n$$

## End