COMP 382: Reasoning about algorithms
Fall 2014

Swarat Chaudhuri

Unit 8: Randomized algorithms
Randomization: Allow fair coin flip in unit time

- In practice, access to pseudorandom number generator

Can lead to the simplest, fastest, or only known efficient algorithm for a particular problem.
Contention resolution

- $n$ processes $P_1, \ldots, P_n$, want to access a shared database
- Time is divided into discrete *rounds*
- If two or more processes access the database simultaneously, all processes are locked out in that round.
Contention resolution

- **Restriction:** Processes cannot communicate
- Devise protocol to ensure all processes get through on a regular basis.
  - Doesn’t make sense for all processes to request access all the time!
Contention resolution

Each process requests access to the database at time $t$ with probability $p = 1/n$
Claim: Let $S[i, t] = \text{event that process } i \text{succeeds in accessing the database at time } t$. Then

$$\frac{1}{e \cdot n} \leq \Pr[S(i, t)] \leq \frac{1}{2n}$$

where $e$ is the base of natural logarithms.
Claim: Let $S[i, t]$ = event that process $i$ succeeds in accessing the database at time $t$. Then $\frac{1}{e \cdot n} \leq \Pr[S(i, t)] \leq \frac{1}{2n}$, where $e$ is the base of natural logarithms.

Proof:

- $S[i, t]$ happens when process $i$ requests access, and none of the other processes request access
- By independence, $\Pr[S(i, t)] = p(1 - p)^{n-1} = \frac{1}{n}(1 - \frac{1}{n})^{n-1}$. 
Claim: Let $S[i, t] = \text{event that process } i \text{ succeeds in accessing the database at time } t$. Then $\frac{1}{e^n} \leq \Pr[S(i, t)] \leq \frac{1}{2n}$, where $e$ is the base of natural logarithms.

Proof:

- $S[i, t]$ happens when process $i$ requests access, and none of the other processes request access.
- By independence, $\Pr[S(i, t)] = p(1-p)^{n-1} = \frac{1}{n}(1 - \frac{1}{n})^{n-1}$.

Useful facts from calculus:

As $n$ increases from 2,
- $(1 - 1/n)^n$ converges monotonically from 1/4 up to 1/e
- $(1 - 1/n)^{n-1}$ converges monotonically from 1/2 down to 1/e.
Analysis

Claim: The probability that process $i$ fails to access the database in $\lceil e \cdot n \rceil$ rounds is at most $1/e$. After $\lceil e \cdot n \rceil \lceil c \ln n \rceil$ rounds, the probability is at most $n^{-c}$. 

Proof: Let $F[i,t]$ = event that process $i$ fails to access database in rounds 1 through $t$. By independence and previous claim, we have $\Pr[F(i,t)] \leq (1 - 1/e)^t$. Choosing $t = \lceil e \cdot n \rceil$: $\Pr[F(i,t)] \leq (1 - 1/e)^{\lceil e \cdot n \rceil} \leq (1 - 1/e)^{e \cdot n} \leq 1/e$. Choosing $t = \lceil e \cdot n \rceil \lceil c \ln n \rceil$: $\Pr[F(i,t)] \leq (1/e)^{c \ln n} = n^{-c}$. 

Swarat Chaudhuri
Claim: The probability that process $i$ fails to access the database in $\lceil e \cdot n \rceil$ rounds is at most $1/e$. After $\lceil e \cdot n \rceil \lceil c \ln n \rceil$ rounds, the probability is at most $n^{-c}$.

Proof: Let $F[i, t] = \text{event that process } i \text{ fails to access database in rounds 1 through } t$. By independence and previous claim, we have $\Pr[F(i, t)] \leq (1 - \frac{1}{en})^t$.

- Choosing $t = \lceil e \cdot n \rceil$:

  $$\Pr[F(i, t)] \leq (1 - \frac{1}{en})^{\lceil en \rceil} \leq (1 - \frac{1}{en})^{\ln n} \leq \frac{1}{e}$$

- Choosing $t = \lceil e \cdot n \rceil \lceil c \ln n \rceil$:

  $$\Pr[F(i, t)] \leq \left(\frac{1}{e}\right)^{c \ln n} = n^{-c}$$
Claim: The probability that all processes succeed within $2e \cdot n \ln n$ rounds is at least $(1 - \frac{1}{n})$.
**Claim:** The probability that all processes succeed within $2e \cdot n \ln n$ rounds is at least $(1 - \frac{1}{n})$.

**Proof:** Let $F[t] =$ event that at least one of the $n$ processes fails to access database in any of the rounds $1$ through $t$. 

$$
\Pr[F[t]] = \Pr[\bigcup_{i=1}^{n} F[i, t]] \leq \sum_{i=1}^{n} \Pr[F[i, t]] \leq n\left(1 - \frac{1}{en}\right)^t
$$

**Union bound**

Given events $E_1, \ldots, E_n$, we have $\Pr[\bigcup_{i=1}^{n} E_i] \leq \sum_{i=1}^{n} \Pr[E_i]$. 

Swarat Chaudhuri  
COMP 382: Reasoning about algorithms
**Claim:** The probability that all processes succeed within $2e \cdot n \ln n$ rounds is at least $(1 - \frac{1}{n})$.

**Proof:** Let $F[t] = \text{event that at least one of the n processes fails to access database in any of the rounds 1 through } t.$

$$
\Pr[F[t]] = \Pr\left[ \bigcup_{i=1}^{n} F[i, t] \right] \leq \sum_{i=1}^{n} \Pr[F[i, t]] \leq n \left(1 - \frac{1}{en}\right)^t
$$

**Union bound**

Given events $E_1, \ldots, E_n$, we have $\Pr[\bigcup_{i=1}^{n} E_i] \leq \sum_{i=1}^{n} \Pr[E_i]$.

Setting $t = 2e \cdot n \ln n$, we have

$$
\Pr[F[t]] \leq n \cdot n^{-2} = \frac{1}{n}.
$$
**Expectation:** Given a discrete random variable $X$, its expectation $E[X]$ is defined by:

$$E[X] = \sum_{j=0}^{\infty} j \cdot Pr[X = j]$$
**Expectation:** Given a discrete random variable $X$, its expectation $E[X]$ is defined by:

$$E[X] = \sum_{j=0}^{\infty} j \cdot Pr[X = j]$$

**Waiting for a first success:** Coin is heads with probability $p$ and tails with probability $1 - p$. How many independent flips $X$ until first heads?
**Expectation**: Given a discrete random variables $X$, its expectation $E[X]$ is defined by:

$$E[X] = \sum_{j=0}^{\infty} j \cdot \Pr[X = j]$$

**Waiting for a first success**: Coin is heads with probability $p$ and tails with probability $1 - p$. How many independent flips $X$ until first heads?

$$E[X] = \sum_{j=0}^{\infty} j \cdot \Pr[X = j] = \sum_{j=0}^{\infty} j(1 - p)^{j-1}p = \frac{p}{1 - p} \sum_{j=0}^{\infty} j (1 - p)^j$$

$$= \frac{p}{1 - p} \cdot \frac{1 - p}{p^2} = \frac{1}{p}$$
**Useful property:** If $X$ is a 0/1 random variable, $E[X] = \Pr[X = 1]$. 
**Useful property:** If $X$ is a 0/1 random variable, $E[X] = \Pr[X = 1]$.  

**Proof:**  

$$E[X] = \sum_{j=0}^{\infty} 1 \cdot \Pr[X = j] = \Pr[X = 1].$$  

**Linearity of expectation:** Given two random variables $X$ and $Y$ defined over the same probability space, $E[X + Y] = E[X] + E[Y]$. 
Guessing Cards

**Game:** Shuffle a deck of $n$ cards; turn them over one at a time; try to guess each card.

**Memoryless guessing:** No psychic abilities; can’t even remember what’s been turned over already. Guess a card from full deck uniformly at random. How often will you be correct?

Claim: The expected number of correct guesses is 1.

Proof using linearity of expectation:

Let $X_i = 1$ if $i$-th prediction is correct and 0 otherwise.

Let $X = \text{number of correct guesses} = X_1 + \cdots + X_n$.

$E[X_i] = \Pr[X_i = 1] = \frac{1}{n}$.

$E[X] = E[X_1] + \cdots + E[X_n] = \frac{1}{n} + \cdots + \frac{1}{n} = 1$. 

Swarat Chaudhuri

COMP 382: Reasoning about algorithms
**Game:** Shuffle a deck of $n$ cards; turn them over one at a time; try to guess each card.

**Memoryless guessing:** No psychic abilities; can’t even remember what’s been turned over already. Guess a card from full deck uniformly at random. How often will you be correct?

**Claim:** The expected number of correct guesses is 1.
**Game:** Shuffle a deck of $n$ cards; turn them over one at a time; try to guess each card.

**Memoryless guessing:** No psychic abilities; can’t even remember what’s been turned over already. Guess a card from full deck uniformly at random. How often will you be correct?

**Claim:** The expected number of correct guesses is 1.

**Proof using linearity of expectation:**

- Let $X_i = 1$ if $i$-th prediction is correct and 0 otherwise.
- Let $X = \text{number of correct guesses} = X_1 + \cdots + X_n$.
- $E[X_i] = \Pr[X_i = 1] = 1/n$.
- $E[X] = E[X_1] + \cdots + E[X_n] = 1/n + \cdots + 1/n = 1$. 
**Game:** Shuffle a deck of $n$ cards; turn them over one at a time; try to guess each card.

**Guessing with memory:** Guess a card uniformly at random from cards not yet seen.
Guessing cards

**Game:** Shuffle a deck of $n$ cards; turn them over one at a time; try to guess each card.

**Guessing with memory:** Guess a card uniformly at random from cards not yet seen.

**Claim:** The expected number of correct guesses is $\Theta(\log n)$. 

Proof using linearity of expectation:

Let $X_i = 1$ if the $i$-th prediction is correct and 0 otherwise.

Let $X = \text{number of correct guesses} = X_1 + \cdots + X_n$.

$E[X_i] = \Pr[X_i = 1] = \frac{1}{n-i-1}$.

$E[X] = E[X_1] + \cdots + E[X_n] = \frac{1}{n} + \cdots + \frac{1}{2} + 1 = H(n)$, where $H(n)$ is the $n$-th harmonic number.

$\ln(n+1) < H(n) < 1 + \ln n$. 

Swarat Chaudhuri

COMP 382: Reasoning about algorithms
**Guessing cards**

**Game:** Shuffle a deck of $n$ cards; turn them over one at a time; try to guess each card.

**Guessing with memory:** Guess a card uniformly at random from cards not yet seen.

**Claim:** The expected number of correct guesses is $\Theta(\log n)$.

**Proof using linearity of expectation:**
- Let $X_i = 1$ if $i$-th prediction is correct and 0 otherwise.
- Let $X = \text{number of correct guesses} = X_1 + \cdots + X_n$.
- $\mathbb{E}[X_i] = \mathbb{P}[X_i = 1] = 1/(n - i - 1)$.
- $\mathbb{E}[X] = \mathbb{E}[X_1] + \cdots + \mathbb{E}[X_n] = 1/n + \cdots + 1/2 + 1/1 = H(n)$, where $H(n)$ is the $n$-th harmonic number
  - $\ln(n + 1) < H(n) < 1 + \ln n$
Exercise: coupon collector

**Coupon collector:**
- Each box of cereal contains a coupon.
- There are $n$ different types of coupons.
- Assuming all boxes are equally likely to contain each coupon, how many boxes before you have $\geq 1$ coupon of each type?
Claim: The expected number of steps is $\Theta(n \log n)$. 

Proof:

Phase $j = \text{time between } j \text{ and } j+1 \text{ distinct coupons}.$

Let $X_j = \text{number of steps you spend in phase } j$.

Probability of success in phase $j$ for each trial is \( \frac{n-j}{n} \), so expected number of steps is 

\[
E[X_j] = \frac{n}{n-j}.
\]

Let $X = \text{number of steps in total} = X_0 + X_1 + \cdots + X_{n-1}$.

\[
E[X] = \sum_{j=0}^{n-1} E[X_j] = \sum_{i=1}^{n} \frac{1}{i} = \sum_{i=1}^{n} \frac{1}{i} = H(n).
\]
**Claim:** The expected number of steps is $\Theta(n \log n)$.

**Proof:**

- Phase $j$ = time between $j$ and $j+1$ distinct coupons.
- Let $X_j =$ number of steps you spend in phase $j$.
- Probability of success in phase $j$ for each trial is $(n - j)/n$, so expected number of steps is $n/(n - j)$
- Let $X =$ number of steps in total $= X_0 + X_1 + \cdots + X_{n-1}$.

\[
E[X] = \sum_{j=0}^{n-1} E[X_j] = \sum_{j=0}^{n-1} \frac{n}{n-j} = n \sum_{i=1}^{n} \frac{1}{i} = nH(n)
\]
Consider a drunken man stumbling through a street lined with street lamps. At each lamp, he independently decides to move right or left: with probability 0.5 he walks left, and with probability 0.5 he moves right. What’s his expected location after $n$ time steps?
Consider a county in which 100,000 people vote in an election. There are only two candidates on the ballot: a Democratic candidate $D$, and a Republican candidate $R$. This county is heavily democratic, so 80,000 people go with the intention of voting for $D$, and 20,000 go with the intention of voting for $R$.

However, the ballot layout is a bit confusing, so each voter, independently and with probability 0.01, votes for the wrong candidate.

Let $X$ denote the random variable equal to the number of votes received by the Democratic candidate $D$. Determine the expected value of $X$. 


RandomizedQuicksort(S) {
    if |S| = 0 return

    choose a splitter a[i] ∈ S uniformly at random
    foreach (a ∈ S) {
        if (a < a[i]) put a in S−
        else if (a > a[i]) put a in S+
    }
    RandomizedQuicksort(S−)
    output a[i]
    RandomizedQuicksort(S+)
}
Quicksort

Running time:
- [Best case] Select the median element as the splitter: quicksort makes $\Theta(n \log n)$ comparisons.
- [Worst case] Select the smallest element as the splitter: quicksort makes $\Theta(n^2)$ comparisons.

Randomize: Protect against worst case by choosing splitter at random.

Intuition: If we always select an element that is bigger than 25% of the elements and smaller than 25% of the elements, then quicksort makes $(n \log n)$ comparisons.

Notation: Label elements so that $x_1 < x_2 < \cdots < x_n$. 
Quicksort: BST representation of splitters

The first splitter, chosen uniformly at random, is represented by the node $x_{10}$.

- $S^-$: Nodes $x_5$, $x_3$, $x_2$, $x_4$, $x_1$, $x_7$, $x_6$, $x_8$
- $S^+$: Nodes $x_{13}$, $x_{11}$, $x_{12}$, $x_{15}$, $x_{14}$, $x_{16}$, $x_{17}$
**Observation:** Element only compared with its ancestors and descendants.

- $x_2$ and $x_7$ are compared if their lca $= x_2$ or $x_7$.
- $x_2$ and $x_7$ are not compared if their lca $= x_3$ or $x_4$ or $x_5$ or $x_6$.

**Claim:** $\Pr[x_i \text{ and } x_j \text{ are compared}] = \frac{2}{j - i + 1}$. 

![BST representation of splitters](image)
Theorem: Expected number of comparisons is $O(n \log n)$.

Proof:

$$\sum_{1 \leq i \leq j \leq n} \frac{2}{j - i + 1} = 2 \sum_{i=1}^{n} \sum_{j=2}^{i} \frac{1}{j} \leq 2n \sum_{j=1}^{n} \frac{1}{j} \leq 2n \int_{x=1}^{n} \frac{1}{x} \, dx = 2n \ln n$$
Theorem: Expected number of comparisons is $O(n \log n)$.

Proof:

$$\sum_{1 \leq i \leq j \leq n} \frac{2}{j - i + 1} = 2 \sum_{i=1}^{n} \sum_{j=2}^{i} \frac{1}{j} \leq 2n \sum_{j=1}^{n} \frac{1}{j} \leq 2n \int_{x=1}^{n} \frac{1}{x} \, dx = 2n \ln n$$

Theorem: [Knuth 1973] Standard deviation of number of comparisons is about 0.65$n$.

- If $n = 1$ million, the probability that randomized quicksort takes less than $4n \ln n$ comparisons is at least 99.94%. 

The sum of independent 0-1 variables is tightly centered on the mean.

**Theorem:** Suppose $X_1, \ldots, X_n$ are independent 0-1 random variables. Let $X = X_1 + \cdots + X_n$. Then for any $\mu \geq \mathbb{E}[X]$ and for any $\delta > 0$, we have

$$\Pr[X > (1 + \delta)\mu] < \left[\frac{e^\delta}{(1 + \delta)^{1+\delta}}\right]^\mu.$$
For any $t > 0$,

$$\Pr[X > (1 + \delta)\mu] = \Pr[e^{tX} > e^{t(1+\delta)\mu}] \leq e^{-t(1+\delta)\mu} \cdot E[e^{tX}]$$

because $\Pr[X > a] \leq E[X]/a$ for all $a$
Proof of Chernoff bounds

- For any $t > 0$,
  \[ \Pr[X > (1 + \delta)\mu] = \Pr[e^{tX} > e^{t(1+\delta)\mu}] \leq e^{-t(1+\delta)\mu} \cdot E[e^{tX}] \]
  because $\Pr[X > a] \leq E[X]/a$ for all $a$

- $E[e^{tX}] = E[e^{t\sum_i X_i}] = \prod_i E[e^{tX_i}]$ due to independence
Proof of Chernoff bounds

For any $t > 0$,

$$\Pr[X > (1 + \delta)\mu] = \Pr[e^{tX} > e^{t(1+\delta)\mu}] \leq e^{-t(1+\delta)\mu} \cdot E[e^{tX}]$$

because $\Pr[X > a] \leq E[X]/a$ for all $a$

$E[e^{tX}] = E[e^{t \sum_i X_i}] = \prod_i E[e^{tX_i}]$ due to independence

Let $p_i = \Pr[X_i = 1]$. Then

$$E[e^{tX_i}] = p_i e^t + (1 - p_i)e^0 = 1 + p_i(e^t - 1) \leq e^{p_i(e^t-1)}$$
Proof of Chernoff bounds

For any $t > 0$,

$$\Pr[X > (1 + \delta)\mu] = \Pr[e^{tX} > e^{t(1+\delta)\mu}] \leq e^{-t(1+\delta)\mu} \cdot E[e^{tX}]$$

because $\Pr[X > a] \leq E[X]/a$ for all $a$

- $E[e^{tX}] = E[e^{t \sum_i X_i}] = \prod_i E[e^{tX_i}]$ due to independence
- Let $p_i = \Pr[X_i = 1]$. Then
  $$E[e^{tX_i}] = p_i e^t + (1 - p) e^0 = 1 + p_i(e^t - 1) \leq e^{p_i(e^t-1)}$$

- Combining everything,
  $$\Pr[X > (1 + \delta)\mu] \leq e^{-t(1+\delta)\mu} \cdot \prod_i E[e^{tX_i}] \leq e^{-t(1+\delta)\mu} \cdot \prod_i e^{p_i(e^t-1)}$$

- Because $\sum_i p_i = E[X] \leq \mu$, $\Pr[X > (1 + \delta)\mu] \leq e^{-t(1+\delta)\mu} \cdot e^{\mu(e^t-1)}$
- We get the result by choosing $t = \ln(1 + \delta)$. 
**Theorem:** Suppose $X_1, \ldots, X_n$ are independent 0-1 random variables. Let $X = X_1 + \cdots + X_n$. Then for any $\mu \leq \mathbb{E}[X]$ and for any $0 < \delta < 1$, we have

$$\Pr[X < (1 - \delta)\mu] < e^{-\delta^2 \mu / 2}.$$ 

**Proof:** Similar.
Consider a drunken man stumbling through a street lined with street lamps. At each lamp, he independently decides to move right or left: with probability 0.5 he walks left, and with probability 0.5 he moves right.

- What’s his expected location after $n$ time steps?
- Give a bound on the probability that he will end up more than $t$ steps away from where he started.
Load balancing: System in which $m$ jobs arrive in a stream and need to be processed immediately on $n$ identical processors. Find an assignment that balances the workload across processors.

Centralized controller: Assign jobs in round-robin manner. Each processor receives at most $\left\lceil \frac{m}{n} \right\rceil$ jobs.
**Load balancing**: System in which $m$ jobs arrive in a stream and need to be processed immediately on $n$ identical processors. Find an assignment that balances the workload across processors.

**Centralized controller**: Assign jobs in round-robin manner. Each processor receives at most $\left\lceil \frac{m}{n} \right\rceil$ jobs.

**Decentralized controller**: Assign jobs to processors uniformly at random. How likely is it that some processor is assigned “too many” jobs?
Load balancing

- Let us consider the case $m = n$.
- Let:
  - $X_i =$ number of jobs assigned to processor $i$
  - Let $Y_{ij} = 1$ if job $j$ assigned to processor $i$, and 0 otherwise
- We have:
  - $\mathbb{E}[Y_{ij}] = 1/n$
  - $X_i = \sum_j Y_{ij}$, and $\mu = \mathbb{E}[X_i] = 1$
Load balancing

- Let us consider the case $m = n$.
- Let:
  - $X_i =$ number of jobs assigned to processor $i$
  - Let $Y_{ij} = 1$ if job $j$ assigned to processor $i$, and 0 otherwise
- We have:
  - $E[Y_{ij}] = 1/n$
  - $X_i = \sum_j Y_{ij}$, and $\mu = E[X_i] = 1$
- Applying Chernoff bounds with $\delta = c - 1$, $Pr[X_i > c] < \frac{e^{c-1}}{c^c}$
- Let $\gamma(n)$ be the $x$ such that $x^x = n$, and let $c = e\gamma(n)$.

\[
Pr[X_i > c] < \frac{e^{c-1}}{c^c} < \left(\frac{e}{c}\right)^c = \left(\frac{1}{\gamma(n)}\right)^{e\gamma(n)} < \left(\frac{1}{\gamma(n)}\right)^{2\gamma(n)} = \frac{1}{n^2}.
\]

- By union bound, with probability $\geq 1 - 1/n$ no processor receives more than $e\gamma(n) = \Theta(\log n / \log \log n)$ jobs.
**Theorem:** Suppose the number of jobs $m = 16n \ln n$. Then on average, each of the $n$ processors handles $\mu = 16 \ln n$ jobs. With high probability every processor will have between half and twice the average load.

**Proof:**

- Let $X_i, Y_{ij}$ be as before.
- Applying Chernoff bounds with $\delta = 1$,

$$\Pr[X_i > 2\mu] < (e/4)^{16n \ln n} < (1/e)^{\ln n} = \frac{1}{n^2}$$

$$\Pr[X_i < \mu/2] < e^{-\frac{1}{2} \left(\frac{1}{2}\right)^2 (16n \ln n)} = \frac{1}{n^2}$$

- By union bound, every processor has load between half and twice the average with probability $\geq 1 - 2/n$. 
Consider an auction system where there are \( n \) bidding agents: agent \( i \) has a bid \( b_i \), which is a positive natural number. We assume that all bids \( b_i \) are distinct from each other.

The bidding agents appear in an order chosen uniformly at random. Each agent proposes its bid \( b_i \) in turn, and at all times the system maintains a bid \( b^* \) equal to the highest bid seen so far. (Initially, \( b^* = 0 \).)

What is the expected number of times that \( b^* \) is updated when this process is executed, as a function of the parameters of the problem?
Consider a situation where you have 2 bins and 2n balls. Each ball independently selects one of the two bins, both bins equally likely. The expected number of balls in each bin is $n$.

Let $X_1$ and $X_2$ be the number of balls in the two bins. Prove that for any $\epsilon > 0$ there is a constant $c > 0$ so that

$$\Pr[X_1 - X_2 > c\sqrt{n}] \leq \epsilon$$