Lecture 7: Intro to Randomized Algorithms

Introduction to Randomized Algorithms

1. A Quick Review of Probability

2. The Hiring Problem

3. Generating a Random Permutation

4. Various Other Items
   - Shuffling Cards
   - The Birthday Paradox
   - Coupon Collectors
   - Generating Random Numbers

A quick review of probability theory

A discrete random variable takes on a countable number of distinct values (usually counts).

Expectation. The *expectation*, $E[X]$, of a discrete random variable X, is defined as:

$$E[X] = \sum i \cdot \Pr[X = i]$$

Q: Roll a 6-sided dice. What is the expected value?

A: $E[X] = \sum_{i=1}^{6} i \cdot \frac{1}{6} = 3.5$

Q: Roll two dice. What is the expected TOTAL value?

A: $E[X] = 2 \cdot \frac{1}{36} + 3 \cdot \frac{2}{36} + 4 \cdot \frac{3}{36} + 5 \cdot \frac{4}{36} + 6 \cdot \frac{5}{36} + 7 \cdot \frac{6}{36} + 8 \cdot \frac{5}{36} + 9 \cdot \frac{4}{36} + 10 \cdot \frac{3}{36} + 11 \cdot \frac{2}{36} + 12 \cdot \frac{1}{36} = 7$

Q: Roll two dice. What is the expected MAXIMUM value seen on a die?

A: $E[X] = 1 \cdot \frac{1}{36} + 2 \cdot \frac{3}{36} + 3 \cdot \frac{5}{36} + 4 \cdot \frac{7}{36} + 5 \cdot \frac{9}{36} + 6 \cdot \frac{11}{36} = 4.47$

A quick review of probability theory (cont)

Expectation. The *expectation*, $E[X]$, of a discrete random variable X, is defined as:

$$E[X] = \sum i \cdot \Pr[X = i]$$

Q (waiting time for the first success): Coin comes up heads with probability p and tails with probability $1 - p$.
How many flips X until first head is seen (lets call it the waiting time, for later use).

A:

$$E[X] = \sum_{j=1}^{\infty} j \cdot Pr[X = j] = \sum_{j=1}^{\infty} j \cdot \underbrace{(1-p)^{j-1}}_{j-1 \text{ tails}} \underbrace{p}_{1 \text{ head}} = p \sum_{j=1}^{\infty} j \cdot (1-p)^{j-1} = p \cdot \frac{1}{p^2} = \frac{1}{p}$$

waiting time$=1/p$

Exercise on a new series

Show that for $0 < p < 1$: $$\sum_{j=1}^{\infty} j(1-p)^{j-1} = \frac{1}{p^2}$$

Hint: use the infinite geometric series, for $0 < x < 1$: $$\sum_{k=0}^{\infty} x^k = 1 + x + x^2 + \ ... = \frac{1}{1-x}$$

The solution is similar to the solution for the geometric series.

Set $x = 1 - p$ $(x < 1)$. Then, $$S = \sum_{j=1}^{\infty} j(1-p)^{j-1} = \sum_{j=1}^{\infty} jx^{j-1} = \sum_{k=0}^{\infty} (k+1)x^k = 1 + 2x + 3x^2 + 4x^3 + \cdots$$

Multiply with x: $$xS = \sum_{j=1}^{\infty} jx^j = \sum_{k=0}^{\infty} (k+1)x^{k+1} = x + 2x^2 + 3x^3 + 4x^4 + \cdots$$

Compute $S - xS$: $$S - xS = (1-x)S = 1 + x + x^2 + x^3 + \cdots = \frac{1}{1-x}$$

Thus: $$S = \frac{1}{(1-x)^2} = \frac{1}{p^2}$$

Linearity of expectation

(1). Given two random variables X and Y (not necessarily independent),

$$E[X+Y] = E[X] + E[Y].$$

Remark: $E[XY] = E[X]E[Y]$ *only when* X *and* Y *are independent.*

Example: Roll two dice. What is the expected TOTAL value X?

We saw that we can calculate

$$E[X] = 2 \cdot \frac{1}{36} + 3 \cdot \frac{2}{36} + 4 \cdot \frac{3}{36} + 5 \cdot \frac{4}{36} + 6 \cdot \frac{5}{36} + 7 \cdot \frac{6}{36} + 8 \cdot \frac{5}{36} + 9 \cdot \frac{4}{36} + 10 \cdot \frac{3}{36} + 11 \cdot \frac{2}{36} + 12 \cdot \frac{1}{36} = 7$$

Easier way is to let X_1, X_2 be the random variables that are values of first and second die. Then $E[X_1] = 3.5, E[X_2] = 3.5$ and by *Linearity of Expectation*

$$E[X_1 + X_2] = E[X_1] + E[X_2] = 3.5 + 3.5 = 7.$$

Exercise Coupon Collector

Coupon collector. Each box of cereal contains a coupon. There are n different types of coupons. Assuming a box contains each type of coupon equally likely, how many boxes do you need to open to have at least one coupon of each type?

Solution.

- Stage i = number of boxes to open between having i and $i+1$ distinct coupons.
- What is the probability p of success at stage 0?
- 1
- What is the probability p of success at stage i?
- $p = \frac{n-i}{n}$
- Let X_i = number of boxes to open at stage i (waiting time at stage i)
- What is the expected value of X_i?
- $\frac{1}{p} = \frac{n}{n-i}$
- What is the expected number of boxes you are expected to open in total?

$$E[X] = \sum_{i=0}^{n-1} E[X_i] = \sum_{i=0}^{n-1} \frac{n}{n-i} = n \sum_{i=1}^{n} \frac{1}{i} = \Theta(n \log n) \qquad (\because \sum_{i=1}^{n} 1/i = \Theta(\log n))$$

Indicator random variables

An indicator random variable X only takes values 0 or 1: $E[X] = \Pr[X = 1]$.

Example. Shuffle a deck of n cards; turn them over one at a time; try to guess each card (n guesses in total). Assume you can't remember what's already been turned over and just guess a card from full deck uniformly at random.

Q. What is the expected number of **correct** guesses?

A. (surprisingly effortless using linearity of expectation)

- Let $X_i = 1$ if i^{th} guess is correct and 0 otherwise.
- Let X = number of correct guesses $= X_1 + \cdots + X_n$.

- $E[X_i] = \Pr[X_i = 1] = 1/n$.
- $E[X] = E[X_1] + \cdots + E[X_n] = 1/n + \cdots + 1/n = 1$.

Exercise on Guessing Cards *with Memory*

Guessing with memory. Shuffle a deck of n cards; turn them over one at a time; try to guess each card. You remember all the cards that have been turned over.

Q. What's the expected number of correct guesses?

A.

- Let $X_i = 1$ if i^{th} guess is correct and 0 otherwise.
- Let X = number of correct guesses = $X_1 + \cdots + X_n$.

- $E[X_i] = \Pr[X_i = 1] = 1/(n-i+1)$.

- $E[X] = E[X_1] + \cdots + E[X_n] = \frac{1}{n} + \cdots + \frac{1}{2} + \frac{1}{1} = \Theta(\log n)$.

Will now use this to analyze a simple RANDOMIZED algorithm

The Hiring Problem

```
Hire-Assistant(n):
best ← 0
for i ← 1 to n
    interview candidate i
    if candidate i is better than best then
        hire candidate i
        best ← i
```

Q: How many people are hired in the worst case and when;

Q: How to avoid the randomness of the input

A: Insert randomness in the algorithm itself

The Hiring Problem: Randomized Algorithm

```
Hire-Assistant(n):
randomly permute all n candidates
best ← 0
for i ← 1 to n
    interview candidate i
    if candidate i is better than best then
        hire candidate i
        best ← i
```

Q: What is the expected number of hires?

A: Similar to *Guessing Cards with Memory*

- Let $X_i = 1$ if you hire candidate i and 0 otherwise.
- Set $X =$ number of hires $= X_1 + \cdots + X_n$.

- $E[X_i] = \Pr[X_i = 1] = 1/i$. (Among the first i candidates, the best has probability $1/i$ to be placed at the last position.)

- $E[X] = E[X_1] + \cdots + E[X_n] = 1 + \frac{1}{2} + \cdots + \frac{1}{n-1} + \frac{1}{n} = \Theta(\log n)$.

Exercise The Birthday Paradox

Problem: Suppose there are $n = 365$ days in a year, and every person's birthday falls on one of the n days with equal probability.
There are k people in a room. How large should k be for us to expect two people in the room to have the same birthday?

Analysis:

- Define $X_{ij} = 1$ if person i and person j have the same birthday, and 0 otherwise.
- We know $E[X_{ij}] = \Pr[X_{ij} = 1] = 1/n$.

- Let $X = \sum_{1 \le i < j \le k} X_{ij}$ be the number of pairs of people having the same birthday.
- We have

$$E[X] = E\left[\sum_{1 \le i < j \le k} X_{ij}\right] = \binom{k}{2}\frac{1}{n} = \frac{k(k-1)}{2n}$$

- So, when $\frac{k(k-1)}{2n} \ge \frac{(k-1)^2}{2n} \ge 1$, or $k \ge \sqrt{2n} + 1 \approx 28$, we expect to see at least one pair of people having the same birthday.

Generating a Random Permutation

- Our solution to Hiring problem required randomly ordering the interview order of the applicants
- Mathematically, we wanted to find a random permutation (ordering of the applicants. How can we do this?
- There are $n!$ different permutations of n items. An algorithm that generates a random permutation would generate each one with probability $1/n!$
- Computers normally only allow you to choose a random integer in a range. How can we use a procedure that generates a random integer to generate a random permutation?
- On next page we assume that our computer has a procedure $Random(1, i)$ that generates a random uniform integer between 1 and i.
  *(**uniform** means that each integer has same probability of occurring)*

How to Generate a Random Permutation

RandomPermute(A):

$n \leftarrow A.length$

for $i \leftarrow 1$ **to** n

  swap $A[i]$ **with** $A[Random(1, i)]$

1	2	3	4	5	6	7	8	9	10	$i = 1$	$Random(1, i) = 1$
2	1	3	4	5	6	7	8	9	10	$i = 2$	$Random(1, i) = 1$
2	3	1	4	5	6	7	8	9	10	$i = 3$	$Random(1, i) = 2$
4	3	1	2	5	6	7	8	9	10	$i = 4$	$Random(1, i) = 1$
4	3	1	2	5	6	7	8	9	10	$i = 5$	$Random(1, i) = 5$
4	6	1	2	5	3	7	8	9	10	$i = 6$	$Random(1, i) = 2$
4	6	1	2	5	3	7	8	9	10	$i = 7$	$Random(1, i) = 7$
4	6	1	8	5	3	7	2	9	10	$i = 8$	$Random(1, i) = 4$
9	6	1	8	5	3	7	2	4	10	$i = 9$	$Random(1, i) = 1$
9	6	1	8	10	3	7	2	4	5	$i = 10$	$Random(1, i) = 5$

How to Generate a Random Permutation-Proof of Correctness

- Precise meaning of a "random permutation":
  Each different permutation is output with probability $1/n!$

- We will show by induction on i that,
  after the i-th iteration, $A[1..i]$ has been randomly permuted,
  - Base case $i = 1$: trivial
  - Assume $A[1..n-1]$= $(a_1, \ldots, a_j, .., a_{n-1})$ has been randomly permuted after $n-1$ iterations of the algorithm.
  - Consider any permutation $(a_1, \ldots, a_n)$ for $A[1..n]$. What's the probability that $A[1..n] = (a_1, \ldots, a_n)$ after the n-th iteration?
  - Let a'_n be the element at position n before the permutation, and $j = Random(1, n)$.
  - Then $(a_1, \ldots, a_n)$ has been created from $(a_1, \ldots, a_j, .., a_{n-1})$ by swapping a'_n with a_j.
  - By the induction hypothesis the probability of producing $(a_1, \ldots, a_j, .., a_{n-1})$ is $1/(n-1)!$.
  - The probability of selecting a specific j is $1/n$.
  - Thus the probability of$(a_1, \ldots, a_n)$ is $1/(n-1)! \cdot 1/n\, 1 = 1/n!$.

How does a computer generate a random number?

Pseudorandom numbers:

- Computed by a deterministic algorithm from a "seed".
- If the "seed" is unknown, then it's difficult to predict the next number to be generated.
  - Often use current machine time as the seed.
- Higher difficulty needs more complicated algorithms.
  - `rand`: "linear generator" $x_n = (214013x_{n-1}+2531011)\ mod\ 2^{32}$
  - `ranlux48`
  - `knuth_b`
  - http://en.cppreference.com/w/cpp/numeric/random

True random numbers:

- Electronic noise, thermal noise, atmospheric noise, etc.
- Expensive and slow
- http://www.random.org

How Humans Do Shuffling

Riffle shuffle

Analysis:

- $\frac{3}{2}\log n$ riffle shuffles can shuffle a deck of n cards to produce a distribution that is close to uniform [Bayer & Diaconis, 1992].
- For $n = 52$, 8 shuffles are good, 7 also OK.