CSCE 411
Design and Analysis of Algorithms

Set 9: Randomized Algorithms
Slides by Prof. Jennifer Welch
Spring 2014
The Hiring Problem

- You need to hire a new employee.
- The headhunter sends you a different applicant every day for $n$ days.
- If the applicant is better than the current employee then fire the current employee and hire the applicant.
- Firing and hiring is expensive.
- How expensive is the whole process?
Hiring Problem: Worst Case

- Worst case is when the headhunter sends you the n applicants in increasing order of goodness.
- Then you hire (and fire) each one in turn: n hires.
Hiring Problem: Best Case

- Best case is when the headhunter sends you the best applicant on the first day.
- Total cost is just 1 (fire and hire once).
Hiring Problem: Average Cost

- What about the "average" cost?
- First, we have to decide what is meant by average.
- An input to the hiring problem is an ordering of the n applicants.
- There are n! different inputs.
- Assume there is some distribution on the inputs
  - for instance, each ordering is equally likely
  - but other distributions are also possible
- Average cost is expected value...
Probability

- Every probabilistic claim ultimately refers to some sample space, which is a set of elementary events.
- Think of each elementary event as the outcome of some experiment.
  - Ex: flipping two coins gives sample space \( \{HH, HT, TH, TT\} \)
- An event is a subset of the sample space.
  - Ex: event "both coins flipped the same" is \( \{HH, TT\} \)
Sample Spaces and Events

<table>
<thead>
<tr>
<th>HH</th>
<th>TH</th>
</tr>
</thead>
<tbody>
<tr>
<td>HT</td>
<td></td>
</tr>
</tbody>
</table>
Probability Distribution

- A probability distribution $Pr$ on a sample space $S$ is a function from events of $S$ to real numbers s.t.
  - $Pr[A] \geq 0$ for every event $A$
  - $Pr[S] = 1$
  - $Pr[A \cup B] = Pr[A] + Pr[B]$ for every two non-intersecting ("mutually exclusive") events $A$ and $B$
- $Pr[A]$ is the probability of event $A$
Probability Distribution

Useful facts:

- \(\Pr[\emptyset] = 0\)
- If \(A \subseteq B\), then \(\Pr[A] \leq \Pr[B]\)
- \(\Pr[S - A] = 1 - \Pr[A]\) // complement
- \(\Pr[A \cup B] = \Pr[A] + \Pr[B] - \Pr[A \cap B]\)
  \(\leq \Pr[A] + \Pr[B]\)
Probability Distribution

\[ \Pr[A \cup B] = \Pr[A] + \Pr[B] - \Pr[A \cap B] \]
Example

- Suppose $Pr[\{HH\}] = Pr[\{HT\}] = Pr[\{TH\}] = Pr[\{TT\}] = 1/4$.
- $Pr[\text{"at least one head"}] = Pr[\{HH \cup HT \cup TH\}] = Pr[\{HH\}] + Pr[\{HT\}] + Pr[\{TH\}] = 3/4$.
- $Pr[\text{"less than one head"}] = 1 - Pr[\text{"at least one head"}] = 1 - 3/4 = 1/4$.
Specific Probability Distribution

- **discrete** probability distribution: sample space is finite or countably infinite
  - Ex: flipping two coins once; flipping one coin infinitely often

- **uniform** probability distribution: sample space $S$ is finite and every elementary event has the same probability, $1/|S|$
  - Ex: flipping two fair coins once
Flipping a Fair Coin

- Suppose we flip a fair coin \( n \) times.
- Each elementary event in the sample space is one sequence of \( n \) heads and tails, describing the outcome of one "experiment".
- The size of the sample space is \( 2^n \).
- Let \( A \) be the event "\( k \) heads and \( n-k \) tails occur".
- \( \Pr[A] = \binom{n}{k}/2^n \).
  - There are \( \binom{n}{k} \) sequences of length \( n \) in which \( k \) heads and \( n-k \) tails occur, and each has probability \( 1/2^n \).
Example

- $n = 5$, $k = 3$
- Event A is
  \{HHHTT, HHTTH, HTTHH, TTHHH, HHTHT, HTHTH, THTHH, HTHHT, THHTH, THHHT\}
- $\Pr[3 \text{ heads and } 2 \text{ tails}] = \frac{C(5,3)}{2^5} = \frac{10}{32}$
Flipping Unfair Coins

- Suppose we flip two coins, each of which gives heads two-thirds of the time.
- What is the probability distribution on the sample space?

\[
\begin{array}{c}
\text{HH} & \text{HT} & \text{TH} & \text{TT} \\
4/9 & 2/9 & 2/9 & 1/9 \\
\end{array}
\]

\[\text{Pr[at least one head]} = 8/9\]
In-Class Problem #1

- What is the sample space associated with rolling two 6-sided dice?
- Assume the dice are fair. What are the probabilities associated with each elementary event in the sample space?
Independent Events

- Two events A and B are independent if
  \[ \Pr[A \cap B] = \Pr[A] \cdot \Pr[B] \]
- I.e., probability that both A and B occur is the product of the separate probabilities that A occurs and that B occurs.
Independent Events Example

In two-coin-flip example with fair coins:

- A = "first coin is heads"
- B = "coins are different"

<table>
<thead>
<tr>
<th>Outcome</th>
<th>Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>HH</td>
<td>1/4</td>
</tr>
<tr>
<td>TH</td>
<td>1/4</td>
</tr>
<tr>
<td>HT</td>
<td>1/4</td>
</tr>
<tr>
<td>TT</td>
<td>1/4</td>
</tr>
</tbody>
</table>

Pr[A] = 1/2
Pr[B] = 1/2
Pr[A \cap B] = 1/4 = (1/2)(1/2)

so A and B are independent
In-Class Problem #2

- In the 2-dice example, consider these two events:
- $A = "\text{first die rolls 6}"$
- $B = "\text{first die is smaller than second die}"$
- Are $A$ and $B$ independent? Explain.
Discrete Random Variables

- A **discrete random variable** $X$ is a function from a finite or countably infinite sample space to the real numbers.
- Associates a real number with each possible outcome of an experiment
- Define the event "$X = v" to be the set of all the elementary events $s$ in the sample space with $X(s) = v$.
- $Pr["X = v"]$ is the sum of $Pr[{s}]$ over all $s$ with $X(s) = v$. 
Discrete Random Variable

Add up the probabilities of all the elementary events in the orange event to get the probability that $X = v$
Random Variable Example

- Roll two fair 6-sided dice.
- Sample space contains 36 elementary events (1:1, 1:2, 1:3, 1:4, 1:5, 1:6, 2:1,...)
- Probability of each elementary event is 1/36
- Define random variable X to be the maximum of the two values rolled
- What is Pr["X = 3"]?
- It is 5/36, since there are 5 elementary events with max value 3 (1:3, 2:3, 3:3, 3:2, and 3:1)
Independent Random Variables

- It is common for more than one random variable to be defined on the same sample space. E.g.:
  - X is maximum value rolled
  - Y is sum of the two values rolled

- Two random variables X and Y are independent if for all v and w, the events "X = v" and "Y = w" are independent.
Expected Value of a Random Variable

Most common summary of a random variable is its "average", weighted by the probabilities called expected value, or expectation, or mean.

Definition: \[ E[X] = \sum_v v \Pr[X = v] \]
Expected Value Example

- Consider a game in which you flip two fair coins.
- You get $3 for each head but lose $2 for each tail.
- What are your expected earnings?
- I.e., what is the expected value of the random variable \( X \), where \( X(\text{HH}) = 6 \), \( X(\text{HT}) = X(\text{TH}) = 1 \), and \( X(\text{TT}) = -4 \)?
- Note that no value other than 6, 1, and -4 can be taken on by \( X \) (e.g., \( \Pr[X = 5] = 0 \)).
- \( \mathbb{E}[X] = 6(\frac{1}{4}) + 1(\frac{1}{4}) + 1(\frac{1}{4}) + (-4)(\frac{1}{4}) = 1 \)
Properties of Expected Values

- \( E[X+Y] = E[X] + E[Y] \), for any two random variables \( X \) and \( Y \), even if they are not independent!
- \( E[a \cdot X] = a \cdot E[X] \), for any random variable \( X \) and any constant \( a \).
- \( E[X \cdot Y] = E[X] \cdot E[Y] \), for any two independent random variables \( X \) and \( Y \).
In-Class Problem #3

- Suppose you roll one fair 6-sided die.
- What is the expected value of the result?
- Be sure to write down the formula for expected value.
Back to the Hiring Problem

- We want to know the expected cost of our hiring algorithm, in terms of how many times we hire an applicant.
- Elementary event $s$ is a sequence of the $n$ applicants.
- Sample space is all $n!$ sequences of applicants.
- Assume uniform distribution, so each sequence is equally likely, i.e., has probability $1/n!$.
- Random variable $X(s)$ is the number of applicants that are hired, given the input sequence $s$.
- What is $E[X]$?
Solving the Hiring Problem

- Break the problem down using indicator random variables and properties of expectation
- Change viewpoint: instead of one random variable that counts how many applicants are hired, consider n random variables, each one keeping track of whether or not a particular applicant is hired.
- Indicator random variable $X_i$ for applicant $i$: 1 if applicant $i$ is hired, 0 otherwise
Indicator Random Variables

- Important fact: \( X = X_1 + X_2 + \ldots + X_n \)
  - number hired is sum of all the indicator r.v.'s

- Important fact about indicator random variables:
  - \( E[X_i] = \Pr[\text{"applicant i is hired"}] \)
  - Why? Plug in definition of expected value.

- Probability of hiring i is probability that i is better than the previous i-1 applicants...
Suppose \( n = 4 \) and \( i = 3 \).

In what fraction of all the inputs is the 3rd applicant better than the 2 previous ones?

\[
\begin{align*}
1234 &\quad 2134 &\quad 3124 &\quad 4123 \\
1243 &\quad 2143 &\quad 3142 &\quad 4132 \\
1324 &\quad 2314 &\quad 3214 &\quad 4213 \\
1342 &\quad 2341 &\quad 3241 &\quad 4231 \\
1423 &\quad 2413 &\quad 3412 &\quad 4312 \\
1432 &\quad 2431 &\quad 3421 &\quad 4321
\end{align*}
\]

\[
8/24 = \frac{1}{3}
\]
Probability of Hiring i-th Applicant

- In general, since all permutations are equally likely, if we only consider the first i applicants, the largest of them is equally likely to occur in each of the i positions.
- Thus $\Pr[X_i = 1] = 1/i$. 
Expected Number of Hires

- Recall that $X$ is a random variable equal to the number of hires.
- Recall that $X = \sum X_i$ (each $X_i$ is the random variable that tells whether or not the $i$-th applicant is hired).
- $E[X] = E[\sum X_i] \quad (i \text{ ranges from } 1 \text{ to } n)$
  $= \sum E[X_i], \text{ by property of } E$
  $= \sum \Pr[X_i = 1], \text{ by property of } X_i$
  $= \sum \frac{1}{i}, \text{ by argument on previous slide}$
  $\leq \ln n + 1, \text{ by formula for harmonic number}$
In-Class Problem #4

- Use indicator random variables to calculate the expected value of the sum of rolling $n$ dice.
Discussion of Hiring Problem

- So average number of hires is $\ln n$, which is much better than worst case number ($n$).
- But this relies on the headhunter sending you the applicants in random order.
- What if you cannot rely on that?
  - maybe headhunter always likes to impress you, by sending you better and better applicants
- If you can get access to the list of applicants in advance, you can create your own randomization, by randomly permuting the list and then interviewing the applicants.
- Move from (passive) probabilistic analysis to (active) randomized algorithm by putting the randomization under your control!
Randomized Algorithms

- Instead of relying on a (perhaps incorrect) assumption that inputs exhibit some distribution, make your own input distribution by, say, permuting the input randomly or taking some other random action.

- On the same input, a randomized algorithm has multiple possible executions.

- No one input elicits worst-case behavior.

- Typically we analyze the average case behavior for the worst possible input.
Randomized Hiring Algorithm

- Suppose we have access to the entire list of candidates in advance
- Randomly permute the candidate list
- Then interview the candidates in this random sequence
- Expected number of hirings/firings is $O(\log n)$ *no matter what the original input is*
Probabilistic Analysis vs. Randomized Algorithm

- Probabilistic analysis of a deterministic algorithm:
  - assume some probability distribution on the inputs

- Randomized algorithm:
  - use random choices in the algorithm
How to Randomly Permute an Array

- input: array A[1..n]
- for i := 1 to n do
  - j := value between i and n chosen with uniform probability (each value equally likely)
  - swap A[i] with A[j]
Why Does It Work?

- Show that after i-th iteration of the for loop:
  \[ A[1..i] \text{ equals each permutation of } i \text{ elements from } \{1,\ldots,n\} \text{ with probability } \frac{1}{n(n-1)\ldots(n-i+1)} \]

- **Basis:** After first iteration, \( A[1] \) contains each permutation of 1 element from \( \{1,\ldots,n\} \) with probability \( \frac{1}{n} \)

- true since \( A[1] \) is swapped with an element drawn from the entire array uniformly at random
Why Does It Work?

- **Induction:** Assume that after the \((i-1)\)-st iteration of the for loop

  \(A[1..i-1]\) equals each permutation of \(i-1\) elements from \(\{1,...,n\}\) with probability

  \[\frac{1}{n(n-1)...(n-(i-1)+1)} = \frac{1}{n(n-1)...(n-i+2)}\]

- The probability that \(A[1..i]\) contains permutation \(x_1, x_2, ..., x_i\) is the probability that \(A[1..i-1]\) contains \(x_1, x_2, ..., x_{i-1}\) after the \((i-1)\)-st iteration AND that the \(i\)-th iteration puts \(x_i\) in \(A[i]\).
Why Does It Work?

- Let $e_1$ be the event that $A[1..i-1]$ contains $x_1, x_2, \ldots, x_{i-1}$ after the $(i-1)$-st iteration.
- Let $e_2$ be the event that the $i$-th iteration puts $x_i$ in $A[i]$.
- We need to show $\Pr[e_1 \cap e_2] = \frac{1}{n(n-1)\ldots(n-i+1)}$.
- Unfortunately, $e_1$ and $e_2$ are not independent: if some element appears in $A[1..i-1]$, then it is not available to appear in $A[i]$.
- We need some more probability…
Conditional Probability

- Formalizes having partial knowledge about the outcome of an experiment
- Example: flip two fair coins.
  - Probability of two heads is 1/4
  - Probability of two heads when you already know that the first coin is a head is 1/2
- Conditional probability of A given that B occurs, denoted Pr[A|B], is defined to be
  \[ \frac{\Pr[A \cap B]}{\Pr[B]} \]
Conditional Probability

\[
\begin{align*}
\Pr[A] &= 5/12 \\
\Pr[B] &= 7/12 \\
\Pr[A \cap B] &= 2/12 \\
\Pr[A | B] &= \frac{2/12}{7/12} = 2/7
\end{align*}
\]
Conditional Probability

- Definition is $\Pr[A \mid B] = \frac{\Pr[A \cap B]}{\Pr[B]}$

- Equivalently, $\Pr[A \cap B] = \Pr[A \mid B] \cdot \Pr[B]$

- Back to analysis of random array permutation...
Why Does It Work?

- Recall: $e_1$ is event that $A[1..i-1] = x_1, ..., x_{i-1}$
- Recall: $e_2$ is event that $A[i] = x_i$
- $\Pr[e_1 \cap e_2] = \Pr[e_2 | e_1] \cdot \Pr[e_1]$
- $\Pr[e_2 | e_1] = 1/(n-i+1)$ because
  - $x_i$ is available in $A[i..n]$ to be chosen since $e_1$ already occurred and did not include $x_i$
  - every element in $A[i..n]$ is equally likely to be chosen
- $\Pr[e_1] = 1/n(n-1)...(n-i+2)$ by inductive hypothesis
- So $\Pr[e_1 \cap e_2] = 1/n(n-1)...(n-i+2)(n-i+1)$
Why Does It Work?

- After the last iteration (the n-th), the inductive hypothesis tells us that A[1..n] equals each permutation of n elements from \{1,...,n\} with probability 1/n!
- Thus the algorithm gives us a uniform random permutation.
Quicksort

- Deterministic quicksort:
  - $\Theta(n^2)$ worst-case running time
  - $\Theta(n \log n)$ average case running time, assuming every input permutation is equally likely

- Randomized quicksort:
  - don't rely on possibly faulty assumption about input distribution
  - instead, randomize!
Randomized Quicksort

- Two approaches
  - One is to randomly permute the input array and then do deterministic quicksort
  - The other is to randomly choose the pivot element at each recursive call
    - called "random sampling"
    - easier to analyze
    - still gives $\Theta(n \log n)$ expected running time
Randomized Quicksort

- Given array $A[1..n]$, call recursive algorithm $\text{RandQuickSort}(A,1,n)$.

- Definition of $\text{RandQuickSort}(A,p,r)$:
  - if $p < r$ then
  - $q := \text{RandPartition}(A,p,r)$
  - $\text{RandQuickSort}(A,p,q-1)$
  - $\text{RandQuickSort}(A,q+1,r)$
Randomized Partition

- RandPartition(A,p,r):
  - i := randomly chosen index between p and r
  - swap A[r] and A[i]
  - return Partition(A,p,r)
Partition

Partition(A,p,r):

- x := A[r]  // the pivot
- i := p–1
- for j := p to r–1 do
  - if A[j] ≤ x then
    - i := i+1
  - swap A[i] and A[j]
- swap A[i+1] and A[r]
- return i+1

A[r]: holds pivot
A[p,i]: holds elts ≤ pivot
A[i+1,j]: holds elts > pivot
A[j+1,r-1]: holds elts not yet processed
Partition

\[
\begin{array}{cccccccc}
    & & & & & & & \\
    i & p,j & r & & & & & \\
2 & 8 & 7 & 1 & 3 & 5 & 6 & 4 \\
p,i & j & r & & & & & \\
2 & 8 & 7 & 1 & 3 & 5 & 6 & 4 \\
p,i & j & r & & & & & \\
2 & 8 & 7 & 1 & 3 & 5 & 6 & 4 \\
p,i & j & r & & & & & \\
2 & 8 & 7 & 1 & 3 & 5 & 6 & 4 \\
\end{array}
\]
Expected Running Time of Randomized QuickSort

- Proportional to number of comparisons done in Partition (comparing current array element against the pivot).
- Compute the expected total number of comparisons done, over all executions of Partition.
- <board work>