Randomization in Algorithms

- **Randomization is a *tool* for designing good algorithms.**
- **Two kinds of algorithms**
  - **Las Vegas - always correct, running time is random.**
  - **Monte Carlo - may return incorrect answers, but running time is deterministic.**

Hiring Problem

$Hire - Assistant(n)$

$best \leftarrow 0$ ▷ **candidate 0 is a least-qualified dummy candidate**
for $i \leftarrow 1$ **to** n
    do interview candidate i
        if candidate i **is better than candidate** $best$
            then $best \leftarrow i$
                hire candidate i

How many times is a new person hired?

Analysis

- A random variable X takes on values from some set, each with a certain probability.
- Expected value: $E[X] = \Sigma_{\textbf{values } x} \Pr(X = x) \cdot x$
- Example: rolling a die.

Expected number of hirings

- **Assume that all orderings of candidates are equally likely.**
- $n!$ **orderings,** $\pi_1, \pi_2, \ldots, \pi_{n!}$
- H **is the total number of hirings.**
- $h(\pi_i)$ **is the number of hirings for permutation** π_i **.**

$$E[H] = \sum_{\pi_i} \frac{1}{n!} h(\pi_i)$$

How do we compute E[H]?

Indicator random variables

- **Let** A **be an event.**
- **The indicator variable** $I\{A\}$ **is defined by:**

$$I\{A\} = \begin{cases} 1 & \textbf{if } A \textbf{ occurs} , \\ 0 & \textbf{if } A \textbf{ does not occur} . \end{cases} \tag{1}$$

What is the expected number of heads when I flip a coin?

- **Let** Y **be a random variable that denotes heads or tails.**
- **Let** X_H **be the i.r.v. that counts the number of heads.**

$$X_H = I\{Y \textbf{ is heads}\} = \begin{cases} 1 & \textbf{if Y is heads} \\ 0 & \textbf{otherwise} \end{cases}$$

$$\begin{aligned} E[X_H] &= \Pr(X_H = 1) \cdot 1 + \Pr(X_H = 0) \cdot 0 \\ &= \frac{1}{2} \cdot 1 + \frac{1}{2} \cdot 0 \\ &= \frac{1}{2} \end{aligned}$$

Linearity of Expectation

Let X and Y be two random variables

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

Linearity of expectation holds even if X and Y are dependent.

n coin flips

- **What is** $E[\textbf{number of heads}]$ **when you flip** n **coins.**
- **Different events are:**
  - **0 heads**
  - **1 head**
  - **2 heads**
  - **3 heads**
  - **. . .**

$$E[\textbf{number of heads}] = \sum_{i=0}^{n} \Pr(\textbf{ i heads in n flips}) \cdot i$$

- **Complicated calculation**
- **Is there another way?**

Use indicator random variables

- **Divide events not by number of heads overall, but by heads in ith flip.**
- **Let X_i be the indicator random variable associated with the event in which the ith flip comes up heads:**
- $X_i = I\{$**the ith flip results in the event** $H\}$**.**
- **Let X be the random variable denoting the total number of heads in the n coin flips**
- $X = \sum_{i=1}^{n} X_i$.
- **We take the expectation of both sides** $E[X] = E[\sum_{i=1}^{n} X_i]$.

$$\begin{aligned} E[X] &= E[\sum_{i=1}^{n} X_i] \\ &= \sum_{i=1}^{n} E[X_i] \\ &= \sum_{i=1}^{n} 1/2 \\ &= n/2 \ . \end{aligned}$$

Hiring

- **Divide events not by number of hires overall, but by hires in ith flip.**
- **Let X_i be the indicator random variable associated with the event in which the ith person is hired**
- $X_i = I\{$**the ith person is hired**$\}$**.**
- **Let X be the random variable denoting the total number of people hired.**
- $X = \sum_{i=1}^{n} X_i$.
- **We take the expectation of both sides** $E[X] = E[\sum_{i=1}^{n} X_i]$.

$$
\begin{aligned}
E[X] &= E[\sum_{i=1}^{n} X_i] \\
&= \sum_{i=1}^{n} E[X_i] \\
&= \sum_{i=1}^{n} \Pr(X_i = 1)
\end{aligned}
$$

What is $\Pr(X_i) = 1$**?**

Analysis

What is $\Pr(X_j = 1)$, the probability that we hire on the j th day?

$$\Pr(X_1 = 1) = ??$$

Analysis

What is $\Pr(X_j = 1)$, the probability that we hire on the j th day?

$$\Pr(X_1 = 1) = 1$$

$$\Pr(X_2 = 1) = ??$$

Analysis

What is $\Pr(X_j = 1)$ **, the probability that we hire on the** j **th day?**

$$\Pr(X_1 = 1) = 1$$

$$\Pr(X_2 = 1) = 1/2$$

$$\Pr(X_j = 1) = ??$$

Analysis

What is $\Pr(X_j = 1)$, the probability that we hire on the j th day?

$$\Pr(X_1 = 1) = 1$$

$$\Pr(X_2 = 1) = 1/2$$

$$\Pr(X_j = 1) = 1/j$$

$$\begin{aligned} E[X] &= E[\sum_{i=1}^{n} X_i] \\ &= \sum_{i=1}^{n} E[X_i] \\ &= \sum_{i=1}^{n} \Pr(X_i = 1) \\ &= \sum_{i=1}^{n} \frac{1}{i} \\ &\approx \ln n \end{aligned}$$

Randomized algorithms vs. Probabilistic Analysis

- **We have assumed that the candidates come in a random order.**
- **Can we remove this assumption?**

Randomized algorithms vs. Probabilistic Analysis

- **We have assumed that the candidates come in a random order.**
- **Can we remove this assumption?**

Randomize the algorithm:

- **Force the candidates to come in a random order by randomly permuting the data, before we start.**
- **We have now eliminated an adversarial-chosen bad case, the only bad case is to be extremely unlucky in our coin flips.**

Case of Sorting

Scenario Imagine a sorting algorithm whose bad case is when the data comes in reverse sorted order.

- **Data is "random":** Bad case is reverse sorted order.
- **Algorithm is random:** some set of coin flips that occur with probability $1/n!$ makes the algorithm slow

Producing a Uniform Random Permutation

Def: **A uniform random permutation is one in which each of the $n!$ possible permutations are equally likely.**

```
RANDOMIZE-IN-PLACE(A)

n ← length[A]
for i ← 1 to n
    do swap A[i] ↔ A[RANDOM(i, n)]
```

Lemma **Procedure RANDOMIZE-IN-PLACE computes a uniform random permutation.**

Def **Given a set of n elements, a k-permutation is a sequence containing k of the n elements.**

There are $n!/(n-k)!$ possible k-permutations of n elements

Proof via Loop invariant

We use the following loop invariant:

Just prior to the ith iteration of the for loop of lines 2– 3, for each possible $(i-1)$-permutation, the subarray $A[1\,..\,i-1]$ contains this $(i-1)$-permutation with probability $(n-i+1)!/n!$.

Initialization

RANDOMIZE-IN-PLACE(**A**)

1 $n \leftarrow length[A]$
2 **for** $i \leftarrow 1$ **to** n
3 **do swap** $A[i] \leftrightarrow A[\text{RANDOM}(i, n)]$

Just prior to the ith iteration of the for loop of lines 2– 3, for each possible $(i-1)$-permutation, the subarray $A[1\,..\,i-1]$ contains this $(i-1)$-permutation with probability $(n-i+1)!/n!$.

Initialization **Consider the situation just before the first loop iteration, so that $i = 1$. The loop invariant says that for each possible 0-permutation, the subarray $A[1\,..\,0]$ contains this 0-permutation with probability $(n-i+1)!/n! = n!/n! = 1$. The subarray $A[1\,..\,0]$ is an empty subarray, and a 0-permutation has no elements. Thus, $A[1\,..\,0]$ contains any 0-permutation with probability 1, and the loop invariant holds prior to the first iteration.**

Maintenance

RANDOMIZE-IN-PLACE(**A**)

```
1  n ← length[A]
2  for i ← 1 to n
3        do swap A[i] ↔ A[RANDOM(i, n)]
```

Just prior to the ith iteration of the for loop of lines 2– 3, for each possible $(i-1)$-permutation, the subarray $A[1\,..\,i-1]$ contains this $(i-1)$-permutation with probability $(n-i+1)!/n!$.

Maintenance **We assume that just before the $(i-1)$st iteration, each possible $(i-1)$-permutation appears in the subarray $A[1\,..\,i-1]$ with probability $(n-i+1)!/n!$, and we will show that after the ith iteration, each possible i-permutation appears in the subarray $A[1\,..\,i]$ with probability $(n-i)!/n!$. Incrementing i for the next iteration will then maintain the loop invariant.**

Let us examine the ith iteration. Consider a particular i-permutation, and denote the elements in it by $< x_1, x_2, \ldots, x_i >$. This permutation consists of an $(i-1)$-permutation $< x_1, \ldots, x_{i-1} >$ followed by the value x_i that the algorithm places in $A[i]$. Let E_1 denote the event in which the first $i-1$ iterations have created the particular $(i-1)$-permutation $< x_1, \ldots, x_{i-1} >$ in $A[1 \mathinner{..} i-1]$. By the loop invariant, $\mathbf{Pr}(E_1) = (n-i+1)!/n!$. Let E_2 be the event that ith iteration puts x_i in position $A[i]$. The i-permutation $< x_1, \ldots, x_i >$ is formed in $A[1 \mathinner{..} i]$ precisely when both E_1 and E_2 occur, and so we wish to compute $\mathbf{Pr}(E_2 \cap E_1)$. Using equation ??, we have

$$\mathbf{Pr}(E_2 \cap E_1) = \mathbf{Pr}(E_2 \mid E_1)\mathbf{Pr}(E_1) \ .$$

The probability $\mathbf{Pr}(E_2 \mid E_1)$ equals $1/(n-i+1)$ because in line 3 the algorithm chooses x_i randomly from the $n-i+1$ values in positions $A[i \mathinner{..} n]$. Thus, we have

$$\begin{aligned}
\mathbf{Pr}(E_2 \cap E_1) &= \mathbf{Pr}(E_2 \mid E_1)\mathbf{Pr}(E_1) \\
&= \frac{1}{n-i+1} \cdot \frac{(n-i+1)!}{n!} \\
&= \frac{(n-i)!}{n!} \ .
\end{aligned}$$

Termination

RANDOMIZE-IN-PLACE(**A**)

```
1   n ← length[A]
2   for i ← 1 to n
3       do swap A[i] ↔ A[RANDOM(i, n)]
```

Just prior to the ith iteration of the for loop of lines 2– 3, for each possible $(i-1)$-permutation, the subarray $A[1\,..\,i-1]$ contains this $(i-1)$-permutation with probability $(n-i+1)!/n!$.

Termination **At termination, $i = n+1$, and we have that the subarray $A[1\,..\,n]$ is a given n-permutation with probability $(n-n)!/n! = 1/n!$.**

Birthday Paradox

Setup:

- n **people**
- **Do two people have the same birthday?**
- **Compute expected number of pairs of people that have the same birthday.**

- X_{ij} **is indicator random variable associated with** i **and** j **having the same birthday.**
- X **is the expected number of pairs that have the same birthday**

$$\begin{aligned} X &= \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} X_{ij} \\ E[X] &= E[\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} X_{ij}] \end{aligned}$$

Birthday Paradox

$$
\begin{aligned}
E[X] &= E[\sum_{i=1}^{n-1}\sum_{j=i+1}^{n} X_{ij}] \\
&= \sum_{i=1}^{n-1}\sum_{j=i+1}^{n} E[X_{ij}] \\
&= \sum_{i=1}^{n-1}\sum_{j=i+1}^{n} \Pr[i \textbf{ and } j \textbf{ have the same birthday}] \\
&= \sum_{i=1}^{n-1}\sum_{j=i+1}^{n} \frac{1}{365} \\
&= \binom{n}{2}\frac{1}{365} \\
&= \frac{n(n-1)}{730}
\end{aligned}
$$

Values

$$
\begin{array}{rr}
n = 23 & .69 \\
n = 28 & 1.03 \\
n = 64 & 5.5 \\
n = 90 & 10.9 \\
n = 140 & 26.6
\end{array}
$$

Streaks

Question: **Suppose we flip n coins, what is the longest streak of heads?**

Answer:

- **Use indicator random variables.**
- **Let X_ik be the event that there is a streak of length k starting at position i. ($A[i \ldots i+k-1]$ are all heads.**
- **Let X_k be the number of streaks of length k.**
- $X_k = \Sigma_{i=1}^{n-k+1} X_{ik}$

$$
\begin{aligned}
E[X_k] &= E[\sum_{i=1}^{n-k+1} X_{ik}] \\
&= \sum_{i=1}^{n-k+1} E[X_{ik}] \\
&= \sum_{i=1}^{n-k+1} \Pr(\textbf{streak of length } k \textbf{ starting at position } i] \\
&= \sum_{i=1}^{n-k+1} 2^{-k} \\
&= \frac{n-k+1}{2^k}
\end{aligned}
$$

What is the behavior of

$$\frac{n-k+1}{2^k}$$

? What is it around 1?

When do we have 1 streak of length k

Think about?

$$n - k + 1 = 2^k$$

so if $k = c \lg n$ **for some** c **, we have**

$$\frac{n-k+1}{2^k} = \frac{n - c\lg n + 1}{2^{c\lg n}} = \frac{n - c\lg n + 1}{n^c}$$

- **if** $c = 1$ **, then the expected number is around 1.**
- **if** $c >> 1$ **, then the expected number starts to decrease rapidly.**
- **if** $c << 1$ **, then the expected number starts to increase rapidly.**
- **so the longest streak should be around length** $\lg n$ **.**