Randomly Permuting Arrays, More Fun with Indicator Random Variables

CS255

Chris Pollett

Feb. 1, 2006.

Outline

- Finishing Up The Hiring Problem
- Randomly Permuting Arrays
- More uses of Indicator Random Variables

Finishing up the Hiring Problem

- Last day we analyzed the Hire-Assistant algorithm assuming the inputs were ordered according to a random uniform permutation.
- We can use coin tosses (hence, a randomized algorithm to ensure this situation).

```
Randomized-Hire-Assistant(n)
    randomly permute the list of candidates
    best <- dummy candidate
    for i <- 1 to n
        do interview of candidate i
            if candidate i is better than best
                then best <- i
                    hire candidate i
```

- So we need a way to generate a random permutation.

Randomly Permuting Arrays (Method 1)

- Idea:
  - start with non-permuted list: *A*=<1,2,3,4>.
  - Generate random priorities: <36,3,97,19>
  - Sort the elements of A according to these priorities to get *B*=<2, 4, 1, 3>
- In more detail:

Permute-By-Sorting(*A*)

1. *n*<-- length[*A*]
2. for *i* <-- 1 to n
3.   do *P*[*i*] = Random(1,n^3)
4. sort *A*, using *P* as sort keys
5. return *A*.

Analyzing Method 1

Lemma: Procedure **Permute-By-Sorting** produces a uniform random permutation of the input, assuming that the priorities are distinct.

Proof: Let σ:[1 .. n] -->[1..n] be a permutation, $\sigma(i)$ being where i goes under this permutation. Let X_i to be the indicator that $A[i]$ receives the $\sigma(i)$th smallest priority. That is, it indicates that i will be mapped correctly after sorting by priorities. So if X_i holds then after sorting the element original value i stored in $A[i]$ gets mapped to $A[\sigma(i)]$. By the definition of conditional probability, $\Pr\{Y|X\} = \Pr\{X \cap Y\}/\Pr\{X\}$, so $\Pr\{X \cap Y\} = \Pr\{X\}*\Pr\{Y|X\}$. Using this, we have $\Pr\{X_1 \cap \ldots \cap X_n\} =$

$\Pr\{X_1 \cap \ldots \cap X_{n-1}\}*\Pr\{X_n \mid X_1 \cap \ldots \cap X_{n-1}\}$. Continuing to expand, we get:

$$Pr\{X_1 \cap X_2 \cap \cdots \cap X_n\} = Pr\{X_1\} \cdot Pr\{X_2|X_1\} \cdots Pr\{X_n|X_{n-1} \cap \cdots \cap X_1\}$$

We can now fill in some of these values:

$\Pr\{X_1\} = 1/n$ = probability that one priority chosen out of n is $\sigma(1)$th smallest.

$\Pr\{X_i|X_1 \cap \ldots \cap X_{i-1}\} = 1/(n - i + 1)$ = since of the remaining elements i, i+1, … n, each is equally likely to be the $\sigma(i)$th smallest.

So $\Pr\{X_1 \cap \ldots \cap X_n\}=1/n*1/(n-1)*\ldots*1/2*1/1 = 1/n!$.

As σ was arbitrary, any permutation is equally likely.

More on Method 1

- What do we do if the priorities aren't all distinct?
- Well, we just try again and draw a new list of priorities.
- What's the likelihood this bad situation happens?

Claim: The probability that all the priorities is unique is at least 1- 1/*n*.

Proof: Let X_i be the indicator that the *i*th priority was unique. Again,

$$Pr\{X_1 \cap X_2 \cap \cdots \cap X_n\} = Pr\{X_1\} \cdot Pr\{X_2 | X_1\} \cdots Pr\{X_n | X_{n-1} \cap \cdots \cap X_1\}$$

$$\begin{aligned}
&= \frac{n^3}{n^3} \frac{n^3 - 1}{n^3} \cdots \frac{n^3 - (n-1)}{n^3} \\
&\geq \frac{n^3 - n}{n^3} \frac{n^3 - n}{n^3} \cdots \frac{n^3 - n}{n^3} \\
&= (1 - \frac{1}{n^2})^{n-1} \geq 1 - \frac{n-1}{n^2} \text{ since } (1-a)(1-b) > (1-a-b) \text{ if and nonnegative} \\
&> 1 - 1/n
\end{aligned}$$

Randomly Permuting Arrays (Method 2)

Randomize-In-Place(*A*)

```
n<-- length[A]
for i <-- 1 to n
      do swap(A[i],  A[Random(i,n)])
```

Analysis of Method 2

Lemma: Just prior to the ith iteration of the for loop, for each possible (i-1)-permutation, the subarray A[1,i-1] contains this permutation with probability $(n-i+1)!/n!$

Proof: By induction on i.

Base case: When i=1, A[1..0] is the empty array. It is supposed to contain a given 0-permutation with probability $(n-1+1)!/n! = n!/n! = 1$. As a 0-permutation has no elements and there is only one of them this is true. For the

Induction step: Assume just before the ith iteration, each (i-1)-permutation occurs in the A[1..i-1] with probability $(n-i+1)!/n!$. A particular, i-permutation $<x_1,\ldots,x_{i-1},x_i>$ consists of an (i-1)-permutation followed by x_i. By the induction hypotheis, the probability of the i-permutation is thus

$$[(n-i+1)!/n!]*\Pr\{A[i]= x_i | A[1..i-1]= <x_1,\ldots,x_{i-1}>\}.$$

The second factor is $1/(n-i+1)$ since by line 3 of Randomize-in-Place, x_i is choosen at random from A[i..n]. So the probability of the i-permutation is $(n-i+1)!/n!*(1/(n-i+1)) = (n-i)!/n!$ as desired.

More Analysis of Method2

Theorem:Randomize-In-Place produces a uniformly chosen random permutation.

Proof: The program could generate any n-permutation. Further it terminates just before its $(n+1)$st iternation and thus by the lemma generates a given random n-permutation with probability:

$(n - (n+1) +1)!/n! = 0!/n! = 1/n!$ as desired.

The Birthday Problem

- How many people must there be in a room before there is a 50% chance that two were born on the same day of the year?
- Let b_1, b_2,.., b_k be IDs for people in the room and their birthday are independent random events.
- Let n be the number of days in a year. (i.e., n=365). Let r be the rth day of year.
- Assume Pr{birthday(b_i) = r}= 1/n.
- Pr{birthday(b_i) = r and birthday(b_j)=r} =

  Pr{birthday(b_i) = r}*Pr{birthday(b_j) = r} = 1/n^2.
- $$\begin{aligned} Pr\{b_i = b_j\} &= \sum_{r=1}^{n} Pr\{\text{birthday}(b_i) = r \text{ and birthday}(b_j) = r\} \\ &= \sum_{i=1}^{n} (1/n^2) \\ &= 1/n. \end{aligned}$$

More on the Birthday Problem

- To determine the odds of whether at least two out of the k people have matching birthday, we look at the complementary event: What are the odds that no-one shares a birthday?
- Let A_i indicate that for no $j<i$, do b_j and b_i have the same birthday.
- Let $B_1=A_1$ and $B_{i+1}=A_{i+1} \cap B_i$.
- So $\Pr\{B_k\} = \Pr\{B_{k-1}\} * \Pr\{A_k | B_{k-1}\}$

$$= \Pr\{B_1\}\Pr\{A_2 | B_1\} * \ldots * \Pr\{A_k | B_{k-1}\}$$

$$= 1\ (1- 1/n)(1- 2/n) \ldots (1 - (k-1)/n)$$

Now can use $1+x <= e^x$ to get this is less than

$e^{-1/n}e^{-2/n} * \ldots * e^{-(k-1)/n} = e^{-(1/n)*(1+2+\ldots+(k-1))} = e^{-k(k-1)/2n}$

which is less than 1/2 if $-k(k-1)/2n <= -\ln 2$. Solving for k using the quadratic formula, this implies $k >= [1+(1+ (8 \ln 2)*n)^{1/2}]/2$. When $n=365$, $k >=23$.