Selected Problems in CLRS

Section 5

Randomized Algorithms are very interesting part. You might learn(have learned) this cheaper in Engineering Mathematics 2.

Notifications

Problem Difficulty (count with star)

1. you can solve w/o the brain
2. you can solve if you think a bit
3. you can solve if you think carefully
4. you might solve if you push yourself
5. you can solve if you use other's brain

Exercise

5.1-2 $\star\star\star$

Describe an implementation of the procedure $\textsc{Random}(a, b)$ that only makes calls to $\textsc{Random}(0, 1)$. What is the expected running time of your procedure, as a function of a and b?.

> **Note :** $\textsc{Random}(a, b)$ is the function that uniformly sample integer $n \in [a, b]$
>
> **Hint1 :** What if $a = 0, b = 2^k - 1$? ($\textsc{Random}(0, 2^k - 1)$)
>
> **Hint2 :** To compute expacted running time, you may use geometric distribution

5.1-3 $\star\star\star$

Suppose that you want to output 0 with probability 1/2 and 1 with probability 1/2. At your disposal is a procedure $\textsc{Biased} - \textsc{Random}$, that output either 0 or 1. It outputs 1 with some probability p and 0 with probability 1-p, where 0 < p < 1, but you do not know what p is. Give an algorithm that uses $\textsc{Biased} - \textsc{Random}$ as a subroutine, and returns an unbiased answer, returning 0 with probability 1/2 and 1 with probability 1/2. What is the expected running time of your algorithm as a function of p?

> **Hint :** You may use two $\textsc{Biased} - \textsc{Random}$ at one iteration. what probability of two events are same?

5.2-4 $\star\star\star$

Use indicator random variables to solve the following problem, which is known as the ***hat-check problem***. Each of n customers gives a hat to a hat-check person at a restaurant. The hat-check person gives the hats back to the customers in a random order. What is the expected number of customers who get back their own hat?

> **Hint :** you can solve easily using linearlity of expactation.

5.2-5 $\star\star\star$

Let $A[1..n]$ be an array of n distinct numbers. If $i < j$ and $A[i] > A[j]$, then the pair (i, j) is called an ***inversion*** of A. Suppose that the elements of A form a uniform random permutation of $\langle 1, 2, ..., n\rangle$. Use indicator random variable to compute the expected number of inversions.

> **Hint1 :** you can solve easily using linearlity of expactation.

> **Hint2 :** If (i, j) is inversion then $I[i, j] = 1$ else $I[i, j] = 0$. What is $\sum I[i, j]$?

5.3-2 $\star\star\star$

Professor Kelp decides to write a prodecure that produces at random any permutation besides the identity permutation. He proposed the following procedure:

```
PERMUTE-WITHOUT_IDENTITY(A)
n = A.length
for i = 1 to n-1
  swap A[i] with A[RANDOM(i+1, n)]
```

> **Hint :** Does this algorithm produce every permutation?

5.3-3 $\star\star\star\star$

Suppose that instead of swapping element $A[i]$ with a random element from the subarray $A[i..n]$, we swapped it with a random element froma anywhere in the array:

```
PERMUTE-WITH_ALL(A)
n = A.length
for i = 1 to n
  swap A[i] with A[RANDOM(1, n)]
```

Does this code produce a uniform random permutation? Why or why not?

> **Note :** If it produces a uniform random permutation, the probability of each permutation is $1/n!$

5.3-5 $\star\star\star\star$

Prove that in the array P in procedure $\textsc{Permute} - \textsc{By} - \textsc{Sorting}$, the probability that all elements are unique is at least $1 - 1/n$

> **Note :** the birthday paradox is similiar with this problem.