Lecture

At the lecture we continue with randomized algorithms. We will talk about random variables, hash tables, and hash functions.

1 [w] Expected values

1.1 Let X be a random variable which assumes the values 2, 5 and 8 with probabilities 1/3, 1/2 and 1/6 respectively, i.e., $\Pr[X = 2] = 1/3$ etc. What is the expected value of X?

1.2 An *indicator random variable* is a random variable that only assumes the values 0 and 1. Prove that for an indicator random variable X we have $E[X] = \Pr[X = 1]$.

2 Analysis of Selection Instead of defining a phase as in the book we now define a phase like follows: The algorithm is in *phase* j when the size of the set under consideration is at most $n(\frac{2}{3})^j$ but greater than $n(\frac{2}{3})^{j+1}$. Do the analysis of Selection with this definition of phase. Do you still get a linear bound on the expected running time?

3 Christmas party at DTU (exam 2015) During the Christmas party at DTU party the Dean suddenly wants to know who won most Christmas cookies in the "Bing or Ding" game. He suggests the following algorithm:

Algorithm 1: Find student with most cookies

```
max ← −∞
s ←null
Randomly order the students. Let s_1,...,s_n be the students in this random order.
Let c_i denote the number of cookies won by student s_i.
for i = 1,...,n do
    if if c_i > max then
        max ← c_i and s ← s_i     (*)
    end
end
return s
```

In the following assume that all students won a different amount of cookies. That is, $c_i \neq c_j$ for all $i \neq j$.

3.1 What is the probability that the line (*) is executed at the last iteration?

3.2 Let X_i be a random variable that is 1 if line (*) is executed in iteration i and 0 otherwise. What is the probability that $X_i = 1$?

3.3 What is the expected number of times line (*) is executed?

4 Nuts and bolts. (G. J. E. Rawlins) You have a mixed pile of N nuts and N bolts and need to quickly find the corresponding pairs of nuts and bolts. Each nut matches exactly one bolt, and each bolt matches exactly one nut. By fitting a nut and bolt together, you can see which is bigger. But it is not possible to directly compare two nuts or two bolts. Given an efficient method for solving the problem.[1]
Hint: customize quicksort to the problem. Side note: only a very complicated deterministic $O(N \log N)$ algorithm is known for this problem.

[1] This exercise is from http://algs4.cs.princeton.edu/23quicksort/

5 Boxes of beer You are given n boxes $B_1, \ldots, B_n$. Exactly k boxes contain a bottle of beer ($k \leq n$) the rest is empty. From the outside one cannot see whether a box is empty or not. The aim is to find a box with a beer it. The following deterministic algorithm is suggested: Open the boxes $B_1, B_2, \ldots$ in this order. The algorithm stops when a beer is found. We count opening a box as one computational step.

5.1 What is the best-case running time of the deterministic algorithm.

5.2 What is the worst-case running time of the deterministic algorithm.

5.3 Consider the following randomized algorithm: Until a beer is found randomly pick a number $i \in \{1, 2, \ldots, n\}$ and open box B_i.

1. What is the expected running time of this algorithm?
2. What is the worst-case running time of this algorithm?

5.4 Now consider a modification of the randomized algorithm above: Until a beer is found randomly pick a box you have not opened before and open it.

We now want to analyse the expected running time of this algorithm. Create an indicator variable X_i for each *empty* box i, where

$$X_i = \begin{cases} 1 & \text{if box } i \text{ was opened by the algorithm} \\ 0 & \text{otherwise.} \end{cases}$$

Let X be the number of boxes opened by the algorithm.

1. Express X using the X_is, i.e., $X = \ldots\ldots$
2. What is the expected value of X_i?
3. What is the expected value of X (use the bounds on the expected value of the X_is)?
4. What is the expected running time of the algorithm?
5. What is the worst case running time of the algorithm?