CMPE 300 – Analysis of Algorithms
Fall 2016
Assignment 2 Answers

Question 1 (50 Points)

Suppose that there are 100 software engineers in a software house. In the first month of the year, a performance evaluation is made and bonus is given to the top 25% of software engineers according to their performance score. The bottom 10% of software engineers get no salary increase. The others get standard salary incrase.

a) I want to find a software engineer who got a bonus payment. Describe a Monte Carlo algorithm for solving the problem which gives a correct answer $\frac{1}{4}$ of the time. What is the runtime of your algorithm? **(15 Points)**

b) I want to find a software engineer who got no salary increase. Describe a Monte Carlo algorithm for solving the problem which gives a correct answer $\frac{19}{100}$ of the time. What is the runtime of your algorithm? **(15 Points)**

c) I want to find a software engineer who got either a standard salary increase or a bonus payment. Describe a Monte Carlo algorithm which gives a correct answer $1 - \left(\frac{10}{100}\right)^k$ of the time for some constant k. What is the runtime of your algorithm? **(20 Points)**

Answer 1

a) Select a software engineer from the company. With probability $\frac{1}{4}$, the selected software engineer got a bonus payment since

$$\frac{number\ of\ software\ engineers\ given\ a\ bonus}{total\ number\ of\ software\ engineers} = \frac{25}{100} = \frac{1}{4}$$

The runtime of the algorithm is O(1).

b) Select a software engineer from the company.
With probability $\frac{1}{10}$, the selected software engineer got no salary increase.
In other words, with probability $\frac{9}{10}$, the selected software engineer either got a standard salary increase or a bonus.
Now select two software engineers and choose the one with the lower performance score.
The answer is not correct if both software engineers either got a standard salary increase or a bonus, which has the probability $\frac{9}{10} * \frac{9}{10} = \frac{81}{100}$.
The correct answer is given with probability $1 - \frac{81}{100} = \frac{19}{100}$.
Only 1 comparison is needed and the runtime of the algorithm is O(1).

c) Select k software engineers sequentially and output the one with the best performance score.
The answer is not correct, if all of the selected k software engineers got no salary increase, which has probability $\left(\frac{10}{100}\right)^k$.
The correct answer is given with probability $1 - \left(\frac{10}{100}\right)^k$.
The runtime of the algorithm is O(1) since the total number of comparisons needed is k-1, which is a constant independent of n.

Question 2 (50 Points)

Suppose you are in a building with k floors and suppose you have an infinite number of identical eggs at your disposal. The eggs you are given break if they are dropped from the N^{th} floor or above. You can drop an egg from a floor in order to test if it breaks. We want to find that N, with minimum number of trials. $N \leq k$ is given. If an egg breaks, it becomes unusable. (**Note:** Eggs do not break if dropped from the 0^{th} floor since they already are on the ground.)

1. Determine the lower bound of 'number of trials to find N' for k floors. **(15 Points)**
2. Now assume you have two eggs instead of infinite number of eggs. Determine the lower bound of 'number of trials to find N' for k floors. Note that you cannot use the same strategy as infinite number of eggs here. You need to be cautious with your eggs so that you find N no matter what. (**Hint:** Think of the case when you have only one egg as a starting point, then analyze the situation based on possible outcomes, and extend it.) **(25 Points)**
3. Solve the problem for ∞ eggs and 1024 floors. **(5 Points)**
4. Solve the problem for 2 eggs and 500 floors. **(5 Points)**

Answer 2

1. First of all, one should note that this is a search problem within the list $\{1, k\}$. We are going to determine the lower bound by using an adversary argument. Let's call the first floor we choose k_1. Then, based on the outcome of the egg dropping action, we are faced with two possible sub-problems: $\{1,\ k_1\}$ or $\{k_1 + 1,\ k\}$, for breaking of egg and survival of egg, respectively. Then, you recursively solve the same problem until you have a single floor. Via this methodology, one can eliminate half of the floors at most at a time. Hence, the lower bound is $log_2 k$.
2. Assume you only have one egg. Then, the only possible solution is to perform a linear search starting from the first floor, since you wouldn't be able to find the answer if you prematurely broke your egg. Note that this is both the worst case and the best case for a single egg.

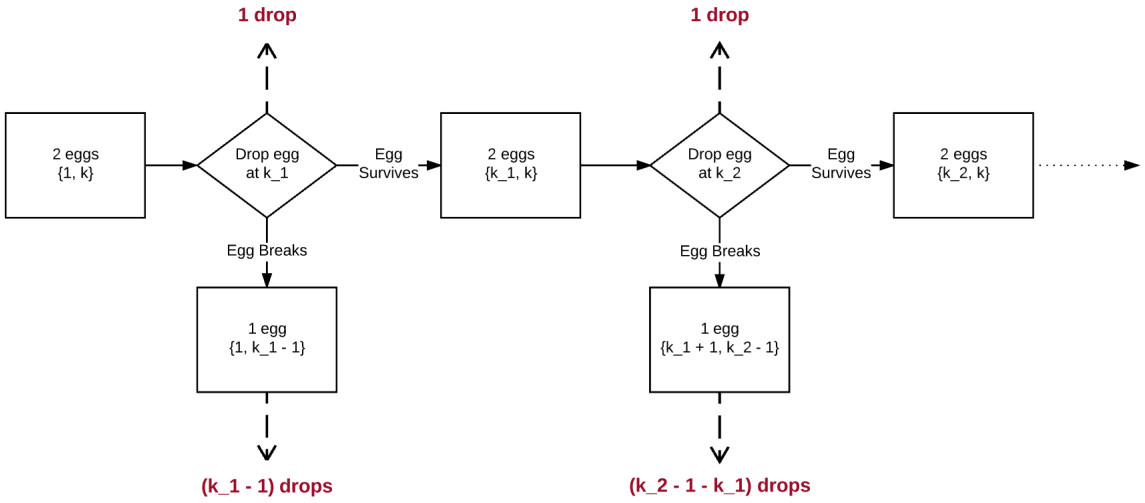
   In this problem, we have two eggs. At any time, we are searching in a sequential list with a lower and upper bound (LB and UB). Let's call the first floor we choose k_1.

   - If the egg breaks, we now have a sub-problem with a single egg within the boundary $\{0,\ k_1 - 1\}$. This is going to take $k_1 - 1$ steps at worst.
   - If the egg does not break, we again have the first problem with two eggs within the boundary $\{k_1 + 1,\ k\}$.

   Now let's look into the second case and call the second floor we choose k_2.

   - If the egg breaks, we now have a sub-problem with a single egg within the boundary $\{k_1 + 1,\ k_2 - 1\}$. This is going to take $k_2 - 1 - k_1$ steps at worst.
   - If the egg does not break, we again have the first problem with two eggs within the boundary $\{k_2 + 1,\ k\}$.

If we were to show this in a flow chart, it would look like this:



This is like a decision tree and since we are performing worst case analysis, we want each path from root to leaves to take the same amount of steps. In a sense the k_i's are the intervals we choose. Between choosing two intervals, a drop happens. Therefore, k_{i+1} can at most be $k_i - 1$. Following this argument, at worst, we shall choose $\{x,\ x+(x-1),\ x+(x-1)+(x-2),\ ...\}$. We keep reducing the step by one each time we choose a new interval, until that increment is just one floor. The increments are as follows, $\{x,\ (x-1),\ (x-2),\ ...,\ ,1\}$. This is a series sum for triangular numbers and it equals to $x*(x+1)/2$. Therefore we shall solve $x*(x+1)/2\ =\ k$, or speaking generally $x*(x+1)/2\ \geq\ k$, since x has to be an integer. x is the worst case lower bound of the problem.

3. $log_2 1024 = 10$
4.

$$\begin{aligned} x*(x+1)/2 &\geq 500 \\ x*(x+1) &\geq 1000 \\ x^2+x-1000 &\geq 0 \end{aligned}$$

The roots for this equation are

$$\begin{aligned} x_1 &= (-1+\sqrt{4001})/2 \\ x_2 &= (-1-\sqrt{4001})/2 \end{aligned}$$

which is approximately

$$\begin{aligned} x_1 &\approx -32.1267 \\ x_2 &\approx 31.1267 \end{aligned}$$

Since x has to be a positive integer, $x = 32$.