CSCI 570: Asymptotic Notation Review/Discussion

Let's review core concepts from the previous lecture. For now, treat this as a "pop quiz" – but don't worry, you aren't being graded! After you have finished answering the questions as you would on a quiz, you are welcome to discuss this with your friends[1] in the class. After some time has been given for everyone to solve and discuss, we will go over solutions before moving onto the new lecture.

1. Suppose we want to determine if a given positive integer $n \geq 2$ is prime (meaning the only numbers that divide it evenly are itself and 1). Here is one algorithm for the problem:

   for $i = 2$ to $n/2$ **do**
     if i divides n evenly **then**
       return false
   return true

   You may assume that arithmetic operations take $O(1)$ time and that this is always called with a valid input value, although note that n is not necessarily a C++ **int** type (it could be more than 32 bits, for example).

   (a) What is the runtime of this algorithm in terms of n, the input value?

   (b) Is this a polynomial time algorithm, as per the definition from lecture and from the reading? Why or why not?

2. Suppose I am going to choose an integer between 1 and n, inclusive, according to some probability distribution. For each integer i, I have written p_i, the probability that I select i as the chosen integer. You may assume that $\sum_{i=1}^{n} p_i = 1$.

   (a) Give an $O(n^3)$ time algorithm to compute a $2D$-array X, where $X[i, j]$ is the probability that some integer in the range $[i, j]$ (inclusive) is chosen. You may assume that arithmetic operations take $O(1)$ time each.
   You should strive to write this in pseudo-code (similar to the algorithm description in the previous problem), although C/C++/Java code is acceptable, as long as the syntax doesn't make it difficult to see the idea behind your code.
   Alternately, if you are certain of your answer to part (b), you may write "See part (b)" for this question; if you do so, part (b) will be graded for both parts (a) and (b).

   (b) Give an $O(n^2)$ time algorithm to solve the problem in part (a).

   There is a problem on the back.

[1] If you don't yet have one or more friends in the class, now is a great time to meet one or more!

3. Suppose we have an array A of n hotels; each hotel has two fields: cost (per night, in dollars) and distance (in miles, representing the distance from the beach). You may assume that each value is distinct (no two hotels cost exactly the same per night, and no two are exactly the same distance from the beach). Our goal is to determine a set of hotels that might satisfy a client looking for the "cheapest hotel that is closest to the beach." Our goal is to report the largest subset of A such that each hotel returned does not have another in A that is simultaneously cheaper and closer to the beach.

   For example, if our dataset is:

Hotel	Cost	Distance from beach
A_1	\$50	3.0 miles
A_2	\$51	5.0 miles
A_3	\$52	4.0 miles
A_4	\$53	2.0 miles

   Then we want to return A_1 and A_4. We don't return A_2 or A_3 because A_1 is both closer to the beach and cheaper.

   (a) Devise an algorithm which takes as input A and n, and outputs the resulting set. Your algorithm does not need to be particularly efficient.

   (b) Analyze the worse-case runtime of your algorithm using Θ-notation.

   (c) Assuming each operation takes 10^{-11} seconds, what is the computing time used by the algorithm, if $n = 250$?. *You may define what counts as a bit operation if it seems ambiguous. Clearly state all assumptions made.*

   (d) Do you believe your algorithm has obtained the best possible asymptotic runtime? Explain your reasoning.