CS302 - Problem Set 6

1. Suppose you have n jobs, and each job i takes time $t_i > 0$ and has a deadline $d_i > 0$. Let the completion time C_i of a job be as defined as in class. Given an ordering of the jobs, we call $l_i = C_i - d_i$ the delay of a job i. We would like to minimize the maximum delay: $\max_j l_j$. Please propose two greedy algorithms that both seem reasonable.

   (a) [**6 points**] Explain why your greedy algorithms are reasonable potential algorithms given the objective.

   (b) [**6 points**] Give a counter example to show that one of them is not optimal.

   (c) [**11 points**] Prove the other is optimal. (Hint: you may want to work backwards from an exchange argument to figure out what the optimal strategy is.) Do NOT assume that there is one unique optimal solution.

   (d) [**3 points**] Briefly describe your algorithm and state its runtime.

2. [**6 points**] Suppose I would like to give you more flexibility on your exams, so I give you some large number (let's call it M) of problems, where the ith problem is worth P_i points. The time I give you to take the exam is not sufficient to solve all of the problems, so each student might solve a different subset of problems (and of course might get different grades on each problem). I would like to give a good grade to a student who does sufficiently well on a sufficient number of questions, and give a worse grade to a student who just gets a few points correct on a lot of problems. What is a (relatively) fair way I could use the knapsack problem to determine grades?

3. Suppose you are given an array of positive numbers $A[1:n]$. You would like to find the largest ratio between two of these numbers, where the numerator occurs after the denominator in the sequence. That is, you would like to compute:

$$\max\left\{\frac{A[i]}{A[j]} : i, j \in \{1, 2, \ldots, n\} \text{ and } i > j\right\}. \tag{1}$$

   See hint on final page if you are getting stuck.

   (Note that proving the algorithm is correct and creating the algorithm are closely intertwined for dynamic programming, so the order of these subproblems is not necessarily the order in which you should solve the problem.)

   (a) [**9 points**] Write pseudocode for a dynamic programming algorithm that finds the maximum ratio.

   (b) [**11 points**] Prove your algorithm is correct.

(c) [**3 points**] What is the runtime of your algorithm?

4. Suppose you have an unordered array A of length n with no repeated elements, and you would like to find the index g such that $A[g] = t$ (so t is the target.) You are promised that t is somewhere in the array. In *random search without replacement*, you choose an index i at random, check whether the element $A[i]$ is t. If it is, return i. If it is not, remove i from the set of indeces you might choose, and repeat by choosing one of the remaining indeces at random. Repeat until t is found. In *random search with replacement*, you do the same process, but you don't remove i from from the set of indeces you might choose in the following round.

   (a) [**3 points**] What is the sample space S for search without replacement if the array A is length 3, and t is the first element of the array? Remember the sample space is the set of all possible sequences of random outcomes over the whole course of the algorithm. (Please list all elements of the sample space.)

   (b) [**3 points**] Let $p : S \to \mathbb{R}$ be the function that gives the probability of each element of S occuring. List $p(s)$ for each $s \in S$.

   (c) [**6 points**] What is the sample space S' for search with replacement if the array A is length 3, and t is the first element of the array? Please describe the set in words or using mathematical notation

   (d) [**6 points**] For search with and without replacement, let the runtime of the algorithms be the number of times the loop is repeated. Describe (using words) the random variables $T : S \to \mathbb{Z}$ and $T : S' \to \mathbb{Z}$ that take in elements of the sample spaces, and output the runtime of the algorithm on that element. In other words, how does the runtime of the algorithm depend on the sequence of random outcomes. Be as specific as possible.

   (e) [**6 points**] What is the average runtime of search without replacement on an array of length 3? That is, you should calculate

   $$\sum_{s \in S} p(s)T(s). \qquad (2)$$

   (f) We can instead use indicator random variables to analyze the average runtime. For example, if $T'(s) = 6$, we would want to think about how we could break T' up into a bunch of indicator random variables, where 6 of the indicator random variables take value 1 on input s, and all of the other indicator random variables take value 0.

      i. [**6 points**] Describe a set of 3 indicator random variables $X_1, X_2, X_3 : S \to \mathbb{Z}$ for search without replacement, such that $X_1(s) + X_2(s) + X_3(s) = T(s)$.

      ii. [**6 points**] Describe a set of infinite indicator random variables X'_i for $i \in \mathbb{N}$ for seach with replacement such that $\sum_{i=1}^{\infty} X'_i(s) = T'(s)$.

   (g) To be continued next week...

5. Approximately how long did you spend on this assignment (round to the nearest hour)?

For Problem 3, you may find it helpful to have your algorithm maintain an array that contains the smallest item yet seen in the array, in addition to an array containing the objective value.