COT5405: ANALYSIS OF ALGORITHMS

Homework # 2

Due date: Feb 14, 2005, Tuesday (beginning of the class)

Your solutions should be concise, but complete, and typed or handwritten clearly. Feel free to consult textbooks, journal and conference papers and also each other, but write the solutions yourself and cite your sources. Answer **only five** of the following six questions. Each problem is worth 20 pts.

1. Given a sequence of n numbers $X = <x_1, x_2, \ldots, x_n>$ a monotonically increasing subsequence is any sequence $X_0 = <x_{a1}, x_{a2}, \ldots, x_{ak}>$ such that $1 \leq a_1 < a_2 < \ldots < a_k \leq n$ and $x_{a1} \leq x_{a2} \leq \ldots \leq x_{ak}$. The goal is to find the length of the longest monotonically increasing subsequence. For example let $X = <1, 4, 3, 5, 17, 10, 15, 12, 20>$, then $X_0 = <1, 3, 5, 10, 15, 20>$ is one of the possible longest monotonically increasing subsequences. The length of this sequence is 6. Note that all the a_i's denote the indices in the sequence X.
   Give an $O(n^2)$ time dynamic programming algorithm to find the longest monotonically increasing subsequence of a sequence of n numbers.

2. An employee working for a company is given n tasks, labeled 1 through n. Each task i requires h_i days to complete, and task i has a deadline d_i in that the task must be finished by the end of day d_i to receive any credit. Answer each of the following parts of the question:

   (a) Suppose all tasks require the same number of days to complete. Describe an algorithm that maximizes the total number of completed tasks within their respective deadlines.

   (b) Suppose all tasks have the same deadline d. Describe an algorithm that maximizes the total number of completed tasks before the deadline.

   (c) Prove that the greedy strategy of Part (a) produces an optimal solution.

3. Given an array $T[1..n]$ containing n arbitrary integer values in a random order. Suppose the (bottom-up) BuildHeap algorithm (discussed in class) is used to convert T into a binary max-heap.

   (a) Suppose $n \geq 3$. Give a pseudocode description of an algorithm that extracts the largest, the second, and the third largest values of the array T. Include the MaxHeapify as one of the steps of your code.

   (b) Consider the case $n = 3$, and the array $T[1..3]$ contains the value 1,2, and 3, in a random order. (There are $3! = 6$ possible cases.) After BuildHeap is applied, determine how many cases out of 6 you find the value 2 (the 2nd largest) is located as $T[2]$. Explain your answer.

   (c) In general, suppose $n \geq 3$ and the original array $T[1..n]$ contains the values 1,2,..n in a random order. What is the probability that after BuildHeap, $T[2]$ = the second largest value of the array? Explain your answer.

4. Pearls have been prized for their beauty and rarity for more than four thousand years. Natural pearls are collected in the coastal regions of Asia, Africa and America. Since pearl collection is dangerous, researchers are asked to build a robot that can do this task. A task for the robot consists of n potential pearl sites all located along a single coastal line. Robot starts at site 1, but can finish at any site. To make things easier we assume that all the sites are on a straight line, one after the other. The time taken to travel from site i to site $i+1$ is $t_{i,i+1}$ hours, $1 \leq i \leq n-1$. Assume that robot's battery lasts for m hours. Also, the expected number of pearls per hour is p_i at site i, $1 \leq i \leq n$. All numbers are nonnegative integers.

   (a) Design an efficient algorithm to find how much time should be spent at each site to maximize the number of pearls found. Give the pseudo-code and run time complexity.

   It was noticed that each hour of pearl collection decreases the expected number of pearls to be found in the next one hour, by a constant amount. This amount varies from site to site and is given as d_i, for site i, $1 \leq i \leq n$.

   (b) Change your algorithm so that it uses the above information to maximize the total pearls found. Give the pseudo-code and run time complexity.

5. A sequence of heavy vehicles are to cross a bridge. It has been found that the bridge can hold not more than W units of weight. Each vehicle has a constant speed s_i and weight w_i, $1 \leq i \leq n$. It is not possible to change the order of the vehicles in the given input sequence. The vehicles can travel in groups (without breaking the sequence) and the speed of the group is the speed of the slowest vehicle in the group. All numbers are nonnegative integers.

   Design an efficient algorithm to divide the sequence of vehicles into groups, so that the time taken for all the vehicles to travel to other side of the bridge is minimized.

6. Suppose instead of powers of two, we represent integers as the sum of Fibonacci numbers. In other words, instead of an array of bits, we keep an array of "fits", where the ith least significant fit indicates whether the sum includes the ith Fibonacci number F_i. For example, the fit string 101110 represents the number $F_6 + F_4 + F_3 + F_2 = 8 + 3 + 2 + 1 = 14$. Describe algorithms to increment and decrement a fit string in constant amortized time. [Note: Most numbers can be represented by more than one fit string.]