NATIONAL UNIVERSITY OF SINGAPORE

SCHOOL OF COMPUTING

SEMESTER I: 2011–2012

EXAMINATION FOR

CS3230 – DESIGN AND ANALYSIS OF ALGORITHMS

November 2011 – Time Allowed 2 Hours

INSTRUCTIONS TO CANDIDATES

1. This examination paper consists of SIX (6) questions and comprises THREE (3) printed pages (including this page).
2. Answer **ALL** questions.
3. This is an **Open Book** examination.

Question 1. (5 marks)

Write in the theta notation the number of times "peace" is printed by the following algorithm on input a positive integer n. Justify your answer.

$print - peace(n)$ {

if $n \geq 1$

  $print - peace(\lfloor n/4 \rfloor)$

  $print - peace(\lfloor n/4 \rfloor)$

for $j = 1$ to $\lfloor \sqrt{n} \rfloor$

  print ("peace")

}

Question 2. (5 marks)

Let the frequencies for alphabets $a_1, a_2, \ldots, a_n$ be $1 \leq f_1 \leq f_2 \leq \cdots \leq f_n$ where $f_n < 2 \cdot f_1$ (here n is a power of 2). What will be the weighted path length of an optimal Huffman tree for these frequencies as a function of $f_1, f_2, \cdots, f_n$ and n? Justify your answer. (Hint : First try special case $n = 4$.)

Question 3. (10 marks)

You are given as input (a, i). Here a is an array (not necessarily sorted) with n positive numbers (not necessarily distinct) each between 1 and m and number i is between 1 and n. You should output the element in a that appears in the ith place in the non-decreasing order. The worst case running time of your algorithm should be $O(m + n)$.

(a) (4 marks) Write idea of the algorithm.

(b) (4 marks) Write pseudocode of the algorithm.

(c) (2 marks) Show that the worst case running time is $O(m + n)$.

Question 4. (10 marks)

You are given n tasks $\{1, \ldots, n\}$ where task i takes time $a[i]$. If a task is started it must be finished. You have to schedule these tasks, one after another in some order, such that the total exit time of all the tasks together is minimized. The exit time of a task is the waiting time plus the time for the task to run. For example let $n = 2$ and there are two tasks $\{1, 2\}$ with $a[1] = 3$ and $a[2] = 5$. If we schedule (this is not necessarily an optimal order) task 2 first and then task 1, the exit time of task 2 is 5 and exit time of task 1 is $5+3 = 8$. Hence total exit time of the two tasks together is $5 + 8 = 13$. Your algorithm is given input array a. You must print the elements of a in the order of an optimal schedule. The worst case running time of your algorithm must be $O(n \log n)$.

(a) (3 marks) Write the idea of the algorithm. (Hint: Use Greedy approach)

(b) (2 marks) Write the pseudocode of the algorithm.

(c) (3 marks) Prove that the algorithm is correct.

(d) (2 marks) Show that the worst case running time of your algorithm is $O(n \log n)$.

Question 5. (10 marks)

There are n balls standing on a line with weights $a[1], a[2], \ldots, a[n]$. You can take any two consecutive balls, say $a[i]$ and $a[i+1]$, and merge them into a bigger ball. The weight of the new ball is $a[i] * a[i+1]$ and the cost of this merging is $a[i] + a[i+1]$. You have to keep doing this till only one ball is left. (Balls cannot be moved form their positions and only two consecutive balls can be merged). For example you are given three balls with weights $3, 1, 2$. If you merge the balls $3, 1$ first, you get two balls with weights $3, 2$ (the cost of this merging is $3 + 1 = 4$) and then you merge these two balls to get one ball of weight 6 (the cost of this merging is $3 + 2 = 5$). Therefore total cost you incur is $4 + 5 = 9$. If you merge the balls $1, 2$ first, you get two balls with weights $3, 2$ (the cost of this merging is $1 + 2 = 3$) and then you merge these two balls to get one ball of weight 6 (the cost of this merging is $3 + 2 = 5$). Therefore total cost you incur is $3 + 5 = 8$.

Write an algorithm, running in time $O(n^3)$, to find the minimum cost you have to incur with best possible way of merging.

(a) (4 marks) Write the idea of the algorithm. (Hint: Use Dynamic programming)

(b) (4 marks) Write the pseudocode of the algorithm.

(c) (2 marks) Show that the worst case running time is $O(n^3)$.

Question 6. (10 marks)

We know 3SAT = $\{< \phi > \mid \phi$ is a satisfiable boolean formula in 3-CNF $\}$.

Let DISJOINT-PATHS =

$\{< G, (s_1, t_1), \ldots, (s_k, t_k) > \mid$ G is a directed graph; $(s_1, t_1), \ldots, (s_k, t_k)$ are pairs of vertices in G and there are vertex-disjoint paths (paths with no common vertices) from s_i to t_i for all $i = 1, \ldots, k$ $\}$

Reduce 3SAT to DISJOINT-PATHS and argue correctness of your reduction. (Hint: The graph that you construct has for each variable two endpoints and two separate parallel paths between them, and for each clause two endpoints connected by three parallel paths that correspond to the literals in the clause. These two kinds of paths should intersect in the appropriate way. You can draw example pictures to illustrate the intersections.)

END of QUESTIONS