Cairo University
Faculty of Computers and Information
Final Exam

Department: CS
Course Name: Algorithms **Date: 2-6-2013**
Course Code: CS316 **Duration: 2 hours**
Instructor(s): Dr Sherif Khattab **Total Marks: 60**
- Dr Basheer Youssef

Part I [20 marks] Attempt all questions

Question 1 [2 marks] Choose the correct answer: ***[1 mark each]***

1. Recall the Partition subroutine used in both QuickSort and RSelect. Suppose that the following array has just been partitioned around some pivot element: 3, 1, 2, 4, 5, 8, 7, 6, 9. Which of these elements could **NOT** have been the pivot element?

a. 5 b. 4 c. 9 d. 8 e. all of the other options

2. Given the following array of ten integers: 5 3 8 9 1 7 0 2 6 4. Suppose we run MergeSort on this array. What is the number in the 7th position of the partially sorted array after the outermost two recursive calls have completed (i.e., just before the very last Merge step)? Assume that counting positions starts at 1.

a. 1 b. 2 c. 3 d. 4 e. none of the other options

Question 2 [7 marks] Answer briefly: ***[1 mark each]***

1. What is the asymptotic worst-case running time of MergeSort, as a function of the input array length n?
2. What is the running time of depth-first search, as a function of n and m, if the input graph G=(V,E) is represented by an adjacency matrix (i.e., NOT an adjacency list), where as usual n=|V| and m=|E| ?
3. Consider a directed graph G=(V,E) with non-negative edge lengths and two distinct vertices s and t of V. Let P denote a shortest path from s to t in G. If we add 10 to the length of every edge in the graph, would P remain a shorting path? why?
4. On adding one extra edge to a directed graph G, would the number of strongly connected components change? why?
5. What is the asymptotic running time of Randomized QuickSort on arrays of length n, in expectation (over the choice of random pivots) **and** in the worst case, respectively?
6. Suppose that the running time of an algorithm is governed by the recurrence $T(n)=7*T(n/2)+n^2$. **What's** the overall asymptotic running time (i.e., the value of T(n))? **Name** an algorithm that has this running time.
7. Recall the Master Method and its three parameters a,b,d. What is the interpretation of b^d in the context of divide-and-conquer algorithms?

Question 3 [5 marks] True or False? correct the wrong statements. ***[1 mark each]***

1. Depth-first search can be used to compute a topological ordering of a directed acyclic graph in O(m+n) time.
2. Depth-first search can be used to compute the strongly connected components of a directed graph in O(m+n) time.
3. Breadth-first search can be used to compute the connected components of an undirected graph in O(m+n) time.
4. Breadth-first search can be used to compute shortest paths in O(m+n) time (when every edge has unit length).
5. Dijkstra's shortest-path algorithm is guaranteed to correctly compute shortest-path distances (from a given source vertex to all other vertices) for input graphs that might have some negative edge lengths.

Question 4 [4 marks] Write an algorithm to solve the "Median Maintenance" algorithm, which is a heap application. The input is a stream of numbers, arriving one by one. Letting x_i denote the *i*th number of the stream, the *k*th median m_k is defined as the median of the numbers $x_1,\ldots,x_k$. (So, if *k* is odd, then m_k is $((k+1)/2)$th smallest number among $x_1, \ldots, x_k$; if k is even, then m_k is the $(k/2)$th smallest number among $x_1, \ldots, x_k$.) **Analyze** the time and space complexities of your algorithm. **Can you** do better if you know that the numbers arrive in sorted order? **how**?

Question 5 [4 marks] Write pseudo-code for **ONLY TWO** of the following:
a. [2 marks] The heap-based implementation of Dijkstra's shortest-path algorithm.
b. [2 marks] An efficient algorithm to compute the connected components of an undirected graph.
c. [2 marks] An efficient algorithm to compute the strongly connected components of a directed graph.

Part II [40 marks]

Question 6 [8 marks]

Write a comparison between Greedy Algorithm vs. Dynamic Programming

Question 7 [8 marks] given the following **Frequencies**

a	50000
b	18000
c	17000
d	21000
e	14000
f	10000

Get the fixed code for each letter and get the Huffman Codes for each letter.

Question 8 [8 marks]

You have to run nine jobs, with running times of 5, 7, 8, 12, 13, 16, 17, 20, and 22 minutes.
You have three processors on which you can run these jobs. Use the greedy algorithm to determine which job run on a specific processor, and compute a total time to run all jobs. Is this solution optimal, why? If not describe how to get optimal one.

Question 9 [8 marks]

Letter	**Probability**
A	0.2
E	0.3
I	0.1
O	0.2
U	0.1
!	0.1

Encode this message “eaii!” using Arithmetic Coding

Question 10 [8 marks]

Write a dynamic programming algorithm to compute a binomial coefficient, and determine the which Data structure used.

Recurrence: $C(n,k) = C(n-1,k) + C(n-1,k-1)$ **for** $n > k > 0$
$C(n,0) = 1, \quad C(n,n) = 1$ **for** $n \geq 0$

What is Time complexity and Space complexity?