

Faculty of Computers and Information

Cairo University

Midterm Exam

|  |  |  |
| --- | --- | --- |
| Department: Special Program SW/NT |  |  |
| Course Name: Algorithms |  | Date: 14 /11/2017 |
| Course Code: | Duration: 1 hour |
| Examiner(s): Dr/ Basheer Youssef | Total Marks: 45 |

**Q1 (5 points)** Arrange the following functions in increasing order of growth rate

a) n2 log(n) b) 2n c) 22^n d) n log(n) e)n2

**Q2 (5 points)** compute the following recurrence

T(1) =50. T(n) = 4T(n/8) +2n for n = 64.

**Q3 (10 points)**

k-way-Merge Sort. Suppose you are given k sorted arrays, each with n elements, and you want to combine them into a single array of kn elements. Consider the following approach. Using the merge subroutine taught in lecture, you merge the first 2 arrays, then merge the 3rdgiven array with this merged version of the first two arrays, then merge the 4th given array with the merged version of the first three arrays, and so on until you merge in the final (kth) input array. What is the running time taken by this successive merging algorithm, as a function of k and n, show details of how to compute the running time.

**Q4 (12 points)** Using master method and substitution method to solve

1. T(n) = 4T(n/2) + n (4 points)
2. T(n) = 4T(n/2) + n2 (4 points)
3. T(n) = 4T(n/2) + n3 (4 points)

**Q5 .(6 points)** For each algorithm listed below give its worst-case running time

1-Binary search 2- Insertion sort 3- Merge sort 4- count sort 5- heap sort 6-redblack tree.

**Q6 (2 points)** what is the main different between skip list and binary search tree? And what is the best method?

**Q7 (5 points)** You are given a unimodal array of n distinct elements, meaning that its entries are in increasing order up until its maximum element, after which its elements are in decreasing order. Give an algorithm to compute the maximum element that runs in   
O (log n) time.