**CS 3130 Design and Analysis of Algorithms Fall 2018**

**T E S T # 1**

**[125 points]**

**----------------------------------------------------------------------------------------------------------**

Name: .

*Important: this is an closed-books/open-notes test. It should be completed within 75 minutes.*

1. [18 points] For each pair of functions shown below, indicate the number of the correct statement:

|  |  |  |  |
| --- | --- | --- | --- |
| Functions | The # of the correct statement | Functions | The # of the correct statement |
| ; |  | ; |  |
| ; |  | ; |  |
| ; |  | ; |  |

1. has the same order of growth as : ;
2. grows slower than : ;
3. grows faster than :.

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

1. [12 points] Circle the number of the correct answer to each of the questions below (there may be multiple correct answers):

|  |  |
| --- | --- |
| A.The worst-case performance is much slower than the average-case performance for …  (1) quicksort; (2) mergesort; (3) selection sort;  (4) insertion sort; (5) none of the above | B.Which of the following algorithms will perform the same number of comparisons for any input?  (1) selection sort; (2) insertion sort;    (3) bubble sort without swaps counting;    (4) bubble sort with swaps counting |
| C.The order of duplicate elements is NOT preserved when the list of elements is sorted with the help of …   1. bubble sort; (2) quicksort; (3) mergesort;   (4) insertion sort; (5) none of the above | D. Sorting algorithms with the quadratic time efficiency are:  (1) quicksort, average case; (2) mergesort;  (3) insertion sort, worst case; (4) quick sort, worst case;  (5)none of the above |

1. [9 points] Use the Master Theorem to determine the order of growth of the solution of each of the following recurrences:

|  |  |  |
| --- | --- | --- |
|  |  |  |

1. [18 points] Given a ***full*** binary tree with 27 internal nodes (it is unknown how many nodes are at each level). Mark each of the following statements as ***T*** (true) or ***F*** (false):

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| This tree has 26 links to external nodes (leaves). |  | The total number of links in this tree is 55 |  | The height of such a binary tree may have the value 3. |  |
| The height of such a binary tree may have the value 12. |  | This tree has 28 external nodes (leaves). |  | The external path length of this tree is 54 greater than the internal path length. |  |

1. [12 points] Given the recurrence: .

Use mathematical induction to show that its solution . Consider only 

It is NOT necessary to check initial conditions. Write your inductive assumption clearly!

1. [9 points] Use the **proper summation formulas** to finish the following calculations:
2. Calculate the sum of 50 terms if the first 4 terms are given as follows:

=

1. Find the infinite sum: =
2. Find the sum: =
3. [10 points] The code fragment implementing the bubble sort (without swaps counting), is shown below. Assuming that *A.length* **=***n*, indicate the number of times each line from this fragment will work, write the analytical expression for the total running time *T*(*n*) (you are supposed to have several summations as a part of it), and ACCURATELY perform all summations.

The result of your calculations should be a function of *n*. When you have this function, determine the order of growth of the running time of a bubble sort *T*(*n*).

|  |  |  |  |
| --- | --- | --- | --- |
|  | BUBBLE-SORT(*A*) | *cost* | *# of times* |
| 1 | **for** *i*=1 **to** *A.length*-1 |  |  |
| 2 | **for** *j* *= A.length* **downto** *i* + 1 |  |  |
| 3 | **if** *A*[*j*] <*A*[*j* -1] |  |  |
| 4 | *exchange* *A*[*j* -1] with *A*[*j*] |  |  |

1. [12 points] Sketch a recursion tree for the recurrence: .

Assuming that *n* is an exact power of 2, show clearly the height of the tree and the number of leaves at the bottom level. Then use the recursion tree to find the asymptotic upper bound of the solution to the given recurrence.

9. An array of integers shown below must be sorted in increasing order.

(a) [9 points] Show the progress of the first call to the partitioning function in a quick-sort algorithm for this array:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | **15** | **6** | **13** | **17** | **14** | **3** | **8** | **5** | **19** | **7** | **20** | **9** | **11** |  |

(b) [6 points] Sort each half of the array after partitioning with the help of **insertion sort** algorithm (on each step show exactly which part of the array is already sorted and how many comparisons were made):

|  |  |  |  |
| --- | --- | --- | --- |
| Progress of insertion sort for the left part of the array after the first partitioning: | | Progress of insertion sort for the right part of the array after the first partitioning: | |
|  | # of  comp. |  | # of  comp. |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
| Total Comparisons: |  | Total Comparisons: |  |

1. (a) [2 points] Give the order of growth of and :
2. [8 points] Rank the functions below from the slowest to the fastest growing.

(1)

*Answer:*

*Probems for extra points:*

A. [5 points] Use the method of ‘telescoping’ (backward substitution) to solve the following recurrence: , .

B. [5 points] Determine how many divisions will be performed by Euclid’s algorithm to find *gcd()*, where **are Fibonacci’s numbers.

C. [5 points] Recall that the depth of a node in a binary tree is its distance from a root. Prove with the help of mathematical induction that there are not more than  nodes at the depth *k* in a binary tree.