Boğaziçi University, Dept. of Computer Engineering

CMPE 250, DATA STRUCTURES AND ALGORITHMS

Fall 2014, Final

Name: ______________________________

Student ID: __________________________

Signature: ___________________________

- Please print your name and student ID number and write your signature to indicate that you accept the University honour code.
- During this examination, you may not use any notes or books.
- Read each question carefully and **WRITE CLEARLY**. Unreadable answers will not get any credit.
- For each question you do not know the answer and leave blank, you can get %10 of the points, if you write only "I don't know the answer but I promise to think about this question and learn its solution".
- There are **7** questions. Point values are given in parentheses.
- You have **180 minutes** to do all the problems.

Q	1	2	3	4	5	6	7	Total
Score								
Max	5	10	15	20	15	15	20	100

1. You observe the following running times for a program with an input of size N.

N	Time
500	0.2 seconds
1000	1.2 seconds
2000	3.9 seconds
4000	16.0 seconds
8000	63.9 seconds

Give a reasonable guess of the complexity as a function of N and estimate the running time of the program (in seconds) on an input of size $N = 20000$.

(5 points)

2. Suppose that you have an array of length $2N$ consisting of N B's followed by N A's. Below is the array when $N = 10$.

   ```
   B B B B B B B B B B A A A A A A A A A A
   ```

   How many compares does it take to insertion sort the array in the increasing order as a function of N? Use asymptotic notation to simplify your answer.

   (10 points)

3. A 3-heap is an array representation of a complete ternary tree, where the key in each node is greater than (or equal to) the keys in each of its children.

   (a) Perform a delete-the-maximum operation on the following 3-heap, which is the level-order traversal of a complete ternary tree, using 1-based indexing (where there is no element at the zero index).

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
–	88	33	77	66	10	30	25	23	60	75	14	21	50	9	7

   Fill in the table below to show the resulting 3-heap, circling any entries that change.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
															–

   (b) Given the array index k of a key, what are the indices of its three (potential) children as a function of k? Assume 1-based indexing and underline your three answers.

   (c) What is the maximum number of compares for a delete-the-maximum operation as a function of the number of keys N in the data structure? Circle the best answer.

   ~ 1 $\sim \log_2 N$ $\sim \log_3 N$ $\sim 2\log_3 N$ $\sim 2\log_2 N$ $\sim 3\log_3 N$ $\sim N$

(15 points)

4. $0-1$ Knapsack problem. Suppose there is a box and N distinct items $k = 1 \dots N$. The k'th item has weight w_k, and has an associated value r_k. We wish to pack some or all of the items into the box such that the total value is maximized, while the box is only allowed to contain items of total weight less than or equal to W. In other words, we wish to

$$\begin{array}{ll} \text{maximize} & \displaystyle\sum_{k=1}^{N} r_k x_k \\ \text{subject to} & \displaystyle\sum_{k=1}^{N} w_k x_k \leq W \end{array}$$

where x_k are decision variables such that $x_k \in \{0, 1\}$, that specify if the item k is chosen or not. Derive an $O(NW)$ Dynamic programming algorithm to find the optimal x_k. You do not have to write the code, it is enough to just show the recursion. Also briefly explain your representation.

For example

k	1	2	3	4	5	6	7
r_k	7	9	5	12	14	6	12
w_k	3	4	2	6	7	3	5

The optimal solution here is

k	1	2	3	4	5	6	7
x_k	1	1	0	0	0	1	1

(20 points)

5. Your job is to arrange N ill-behaved children in a straight line, facing front. You are given a list of M statements of the form i hates j. If i hates j, then you do not want to put i somewhere behind j, because then i is capable of throwing something at j. Give an algorithm that orders the line, (or says that it is not possible) in $O(N + M)$ time.

(15 points)

6. Fill in the following table. (Leave empty if you are unsure as a wrong answer cancels one right answer)

	Insertion Sort	Heapsort	Mergesort	Quicksort
Sequence num				

Below, the column on the left is the original input of strings to be sorted; the column on the right are the string in sorted order; the other columns are the contents at some intermediate step during one of 7 sorting algorithms some of which are listed above. Match up each column by writing its number to the corresponding row labeled as 'sequence'. Use each number exactly once. Briefly, justify your answer.

U	1	2	3	4	5	6	7	S
25	1	22	32	1	13	1	4	1
17	2	17	30	4	15	2	6	2
13	4	13	31	6	17	3	8	3
15	3	15	28	8	25	4	11	4
6	6	6	27	9	6	5	13	5
11	5	11	29	11	8	6	14	6
8	8	8	26	13	11	7	15	7
14	12	14	18	14	14	8	17	8
28	7	18	16	15	4	9	20	9
27	15	19	23	17	20	10	25	10
4	9	4	24	20	27	11	27	11
20	14	20	25	21	28	12	28	12
1	10	1	22	25	1	25	1	13
21	16	21	21	26	9	21	21	14
26	11	7	19	27	21	26	26	15
9	18	9	14	28	26	28	9	16
3	13	3	3	2	2	13	3	17
16	22	16	15	3	3	16	16	18
2	17	2	2	10	16	17	2	19
30	20	12	6	16	30	30	30	20
23	21	23	17	23	10	23	23	21
10	24	10	10	24	23	27	10	22
24	19	24	4	30	24	24	24	23
31	26	5	20	31	31	31	31	24
29	23	25	13	29	5	29	29	25
32	28	32	1	32	22	32	32	26
22	25	29	11	22	29	22	22	27
5	30	31	5	5	32	15	5	28
12	29	30	12	12	7	20	12	29
7	32	26	7	7	12	14	7	30
19	27	27	8	19	18	19	19	31
18	31	28	9	18	19	18	18	32

(15 points)

7. Give an $O(N \log K)$ time algorithm that merges K sorted lists with a total of N elements into a single sorted list. Describe the algorithm but do not write pseudocode. Also describe why your algorithm runs in $O(N \log K)$ time.
   (Hint: Use a heap to speed up the elementary $O(KN)$ time algorithm)

   (20 points)