NATIONAL UNIVERSITY OF SINGAPORE

SCHOOL OF COMPUTING
WRITTEN QUIZ 1 (14%)
AY2017/18 Semester 1

CS2010 – Data Structures and Algorithms II

22 September 2017 Time allowed: 90 minutes

INSTRUCTIONS TO CANDIDATES

1. Do **NOT** open the question paper until you are told to do so.

2. This question paper contains THREE (3) sections with sub-questions. Each section has a different length and different number of sub-questions. It comprises TEN (10) printed pages, including this page.

3. Answer all questions in this paper itself (write in the empty space provided). You can use either pen or pencil. Write **legibly**!

4. This is an **Open Book Quiz**. You can check the lecture notes, tutorial files, problem set files, CP3 book, or any other books that you think will be useful. But remember that the more time that you spend flipping through your files implies that you have less time to actually answer the questions.

5. When this Quiz starts, **immediately first write your Matric Number** and **Tutorial Group**.

6. The total marks for this paper is **50**.

TUTORIAL GROUP

STUDENT NUMBER: A | | | | | | | | |

For examiners' use only		
Question	***Max***	***Marks***
Q1-4	12	
Q5	8	
Q6	10	
Q7	10	
Q8	10	
Total	**50**	

Section A – Analysis (12 Marks)

Prove (the statement is correct) or disprove (the statement is wrong) the following statements below. If you want to prove it, provide the proof or at least a convincing argument. If you want to disprove it, provide at least one counter example. 3 marks per each statement below (1 mark for saying correct/wrong, 2 marks for explanation):

1. Instead of using a 1-based compact array to store a binary heap, we can use a 0-based compact array if we modify the navigation operations as follows
   - `parent(i) = i/2  if i > 0, undefined otherwise`
   - `left(i) = (i*2)+1  if i < heap size, undefined otherwise`
   - `right(i) = (i*2)+2  if I < heap size, undefined otherwise`

2. The smallest AVL tree where deleting any vertex will cause 2 re-balancing operations (the 4 cases given in lecture) has 13 vertices.

3. To remove an item i from a disjoint set that has ≥ 2 items in a UFDS and make it the only item of a new set, we can simply set P[i] = i.

4. Assume that ShiftDown(i) will always fix max heap property of the subtree rooted at i iff i is the only node that might violate max heap property. Fast heap creation for a max heap of size N will still work if we modify it as follows

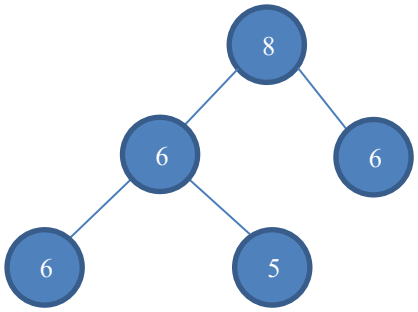
```
For i = floor(logN) down to 0
  For j = 2^i to min(heapsize,2^(i+1)-1)
    ShiftDown(j)
```

Section B – Extensions (18 Marks)

Write in pseudo-code. However, any algorithm/functions not taught in CS2010/CS1020 must be described, there must be no black boxes. **Partial marks will be awarded for correct answers not meeting the time complexity required**. (* questions might be harder).

5. **New max heap operation [8 marks]**

For a max heap containing possibly repeated integer keys, write an algorithm for a new operation **int frequency(i)** which returns the number of items in the heap with the same key as the item at index i. **frequency(i)** must run in time $O(1)$. Insert and ExtractMax operation must be maintained at $O(logN)$ respectively. For example, given the max heap below



If frequency(2) is called, then it should return 3.

In your answer, describe any auxiliary data structure you require and also any modification to Insert and ExtractMax (and argue they still maintain O(logN) runtime).

6. **More BST/bBST operations [10 marks]**

i.) Given T the root of a BST containing unique integer values (no repeated values) of size N. Write function **boolean isAVL(*T*)** which will return true if T is also an AVL tree in $O(N)$ time. You cannot assume the height attribute is already computed for each vertex. **[5 marks]**

ii.) Given roots of two **BST** T and T' of size N and M respectively, containing unique integers, write the algorithm for a merge operation which will merge the two BST into an **AVL tree** (and return the root of the new tree) in O(N+M) time. **[*5 marks]**

Section C – Application (20 Marks)

Write in pseudo-code. However, any algorithm/functions not taught in CS2010/CS1020 must be described, there must be no black boxes. **Partial marks will be awarded for correct answers not meeting the time complexity required.** (* questions might be harder).

7. **A range of values [10 marks]**

Given an array of N unsorted and non-repeated integers, output the M smaller and the M larger values than the median of the N integers, including the median. They must be output in ascending order. Assume that $M \geq 1$, N is odd and $N \leq 1{,}000{,}000{,}000$, and $2M$ is much smaller than N ($2M \ll N$). Your algorithm must run in expected time O($N \log M$) or better.

For example, given [132,13,151,15,1,41,6] the median is 15. If M=2, then 6,13,15,41,132 will be output.

Solve the problem using any data structure you have learned in CS1020 & CS2010. **[10 marks]**

8. **Rebuild the code! [10 marks]**

Jane the manager of a bank has the code for the safe on a piece of paper that is locked in her office drawer. One day, she discovers her office ransacked and the paper missing! However, Jane does not panic because she realizes it is not easy for the thief to use the code to open the bank safe.

This is because the bank safe code is very special. It consists of N integers ($2 \leq N \leq 1{,}000{,}000$). This is not the special thing about the code. The special thing is that they are actually arranged in a BST and this BST is the one that is to be entered into the security system in order to open the safe (Imagine there is some way to do it …).

What was written on the piece of paper is simply a sequence of the N integers obtained by either the **in-order**, **pre-order** or **post-order** traversal of the BST.

But wait! Jane remembers that she has put an asterisk beside the integer which is the root of the BST. This could provide vital information for the thief to rebuild the BST!

You are the thief who stole the piece of paper, and you have entered the sequence of N integers into an array A. Answer the following questions in order to reconstruct the BST.

i.) You realize that in general it is not possible for the sequence to be generated using in-order traversal. Argue why this is so (Do this like an analysis question). **[3 marks]**

ii.) Now given the array containing the sequence A and the index of the root r in A, how do you tell whether it is a pre-order or post-order sequence? **[2 marks]**

iii.) Now assuming that the sequence is generated using post-order traversal, write the algorithm for **T RebuildTreePost(A)** which will rebuild the BST in $O(N)$ time given A the array containing the sequence and return T the root the rebuilt tree. You can include any additional helper functions you need. **[*5 marks]**

== END OF PAPER ==