National University of Singapore

School of Computing

Semester 1 (2015/2016)

CS2010 - Data Structures and Algorithms II

Written Quiz 1 (10%)

Saturday, September 19, 2015, 10.00am-11.30am (90 minutes)

INSTRUCTIONS TO CANDIDATES:

1. Do **NOT** open this question paper until you are told to do so.
2. Written Quiz 1 is conducted at 3 adjacent venues: LT19 ['A'..'P'] (111 students, capacity 200), SR@LT19 ['Q'..'U'] (37 students, capacity 42), and TR9 ['W'..'Z'] (31 students, capacity 32).
3. This question paper contains THREE (3) sections with sub-questions.
   It comprises EIGHT (8) printed pages, including this page.
4. Write all your answers in this question paper, **but only in the space provided**.
   You can use either pen or pencil. Just make sure that you write **legibly**!
   Important tips: Pace yourself! Do **not** spend too much time on one (hard) question.
5. This is an **Open Book Examination**. You can check the lecture notes, tutorial files, problem set files, Steven's 'Competitive Programming 1/2/2.5/3' book, or any other printed material 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 answering the questions.
6. **Please write your Matriculation Number here:** _____ **and your Tutorial Group Number/Tutor Name/Monday or Tuesday:** _____
   But do not write your name in order to facilitate unbiased grading.
7. All the best :).
   After Written Quiz 1, this question paper will be collected, graded manually over recess week, and likely returned to you via your Tutor on Week07.

---

1 Analysis (15 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 (preferred) or at least a convincing argument.
If you want to disprove it, provide at least one counter example.
Three marks per each statement below (1 mark for saying correct/wrong, 2 marks for explanation):
Note: You are only given a small amount of space below (i.e. do **not** write too long-winded answer)!

1. In `ShiftDown(i)` operation of Binary (Max) Heap, there is this pseudocode: "if A[i] $<$ than the larger of its children, swap A[i] with that child". Professor ABC claims that it is OK to change this pseudocode into: "if A[i] $<$ than any of its children, swap A[i] with that child".

2. There is an AVL Tree with $n = 1000$ vertices that has height $h = 30$.

3. The 'union-by-rank' and 'path-compression' heuristics of the Union-Find Disjoint Sets data structure are not that important. We can still achieve the $O(1)$ performance if we are only dealing with up to 1M (1 Million) items.

4. An integer is stored in binary in computer memory. Therefore we can use this sequence of 0s and 1s to represent a medium-size set of **10000** Booleans very efficiently and can manipulate that integer via various $O(1)$ bit manipulation operations (e.g. '&', '|', '<<', '>>').

5. An adjacency matrix representation M of a graph G is symmetrical, i.e. $M = M^T$. Therefore, graph G must be an Undirected Graph.

2 Not Yet in VisuAlgo Online Quiz (50 marks)

2.1 `Extract(i)` Method of a Binary (Max) Heap (10 marks)

In PS1, you were introduced with the `Extract(i)` method of a Binary (Max) Heap (disguised as `GiveBirth(womanName)` method). One possible implementation of this method is:

```
Extract(i) // i is the index of the element to be extracted
  swap(A[i], A[heapsize])
  heapsize--;
  ShiftUp(i);   // line 3, fix potential (Max) Heap property violation upwards
  ShiftDown(i); // line 4, fix potential (Max) Heap property violation downwards
```

Figure 1: The Starting Binary Max Heap

Now your task is this: Given Binary Heap structure of $n = 7$ integers: {70, 30, 60, 10, 20, 40, 50} as shown in Figure 1 (note that the root 70 is at index 1, not at index 0), list down all indices $\in$ [1..7] so that if `Extract(i)` is called:

1. (3 marks) `ShiftUp(i)` in Line 3 triggers some swaps but `ShiftDown(i)` in Line 4 does not
   My answer: _____
2. (2 marks) `ShiftDown(i)` in Line 4 triggers some swaps but `ShiftUp(i)` in Line 3 does not
   My answer: _____
3. (3 marks) Neither `ShiftUp(i)` in Line 3 nor `ShiftDown(i)` triggers any swap
   My answer: _____
4. (2 marks) Both `ShiftUp(i)` in Line 3 and `ShiftDown(i)` trigger some swaps
   My answer: _____

2.2 AVL Tree Insertion Balancing Act Challenge (10 marks)

Give a sequence of insertion of this set of $n = 7$ integers {1, 2, 3, 4, 5, 6, 7} into an initially empty AVL tree so that **no** rotation occurs at all. Please explain why that insertion order does not trigger any rotation.

2.3 AVL Tree Deletion Domino Effect Challenge (10 marks)

In Lecture04 and tut03, you have been presented with this example (see Figure 2) where deleting vertex 7 causes two groups of rotations to be performed, that is: rotateRight(6)—the first group followed by rotateRight(16)+rotateLeft(8)—the second group. Notice that the second group consists of two individual rotations but counted as one group.

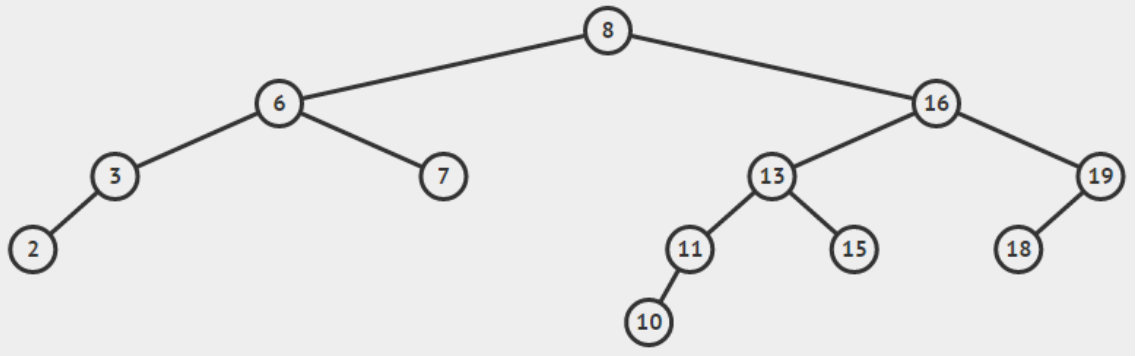

Figure 2: The Example AVL Tree

Now your task is simply this: Draw **any** AVL tree structure of n integers with value $[1..n]$, nominate a vertex between that range $[1..n]$ so that if that vertex is deleted, **THREE** groups of rotation(s) happen. To facilitate easier grading, mention what are those three groups of rotation(s) are.

2.4 UDFS Min/Max Tree Height Challenge (20 marks)

In Lecture05, you have learned about Union-Find Disjoint Sets (UFDS) data structure that uses both 'union-by-rank' and 'path-compression' heuristics. Given an UFDS structure that currently has 4 disjoint sets as shown in Figure 3 below.

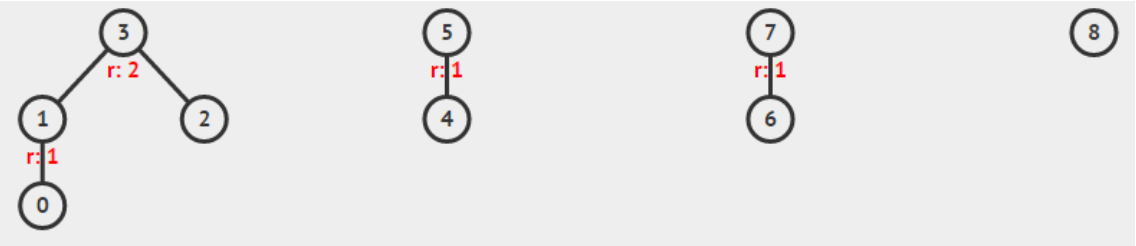

Figure 3: The current UFDS structure with 4 disjoint sets

Your task is to **perform exactly three** `unionSet(i, j)` operations so that there is only one set left and **draw the resulting tree**. There are two conditions that you have to fulfill.

1. (10 marks) The resulting tree is as tall as possible
   My answer: Do `unionSet(____, ____)`, `unionSet(____, ____)`, and then `unionSet(____, ____)`
   My resulting tree:

2. (10 marks) The resulting tree is as short as possible
   My answer: Do `unionSet(____, ____)`, `unionSet(____, ____)`, and then `unionSet(____, ____)`
   My resulting tree:

3 Think Outside the Box (35 marks)

3.1 CS2010 Leaderboard (20 marks)

You must have visited URL: `http://www.comp.nus.edu.sg/~stevenha/cs2010.html#leaderboard` at least once during this semester and possibly many more times (maybe every week) to monitor where you are in class :).

This leaderboard can be abstracted as a data structure that contains n triples of information: `(studentname, onlinequizscore, psscore)`. Two of the three fields `(onlinequizscore, psscore)` are **frequently updated** throughout the semester whereas field `studentname` is static. Consider that CS2010 is offered to worldwide audience (it is a MOOC – Massive Open Online Course) so that there are $n = 999999$ students enrolled (that is, 1 Million minus 1 students), an odd number. Steven wants to build an efficient leaderboard that can perform all these methods efficiently:

1. `UpdateScore(studentname, newonlinequizscore, newpsscore)`
   As the method name implies, this method updates the `onlinequiz` and `psscore` of `studentname` to new values. If `studentname` does not exist before, we create a new triple.

2. `GetBest(type)`
   This method returns the `studentname` who has the best score, depending on attribute type:

   (a) if `type = 0`, return the best `studentname` with the highest `onlinequizscore`

   (b) if `type = 1`, return the best `studentname` with the highest `psscore`

   (c) if `type = 2`, return the best `studentname` with the highest `(onlinequizscore+psscore)`

3. `GetMedian(type)`
   This method returns the `studentname` who has the median score, depending on attribute type:

   (a) if `type = 0`, return the best `studentname` with the median `onlinequizscore`

   (b) if `type = 1`, return the best `studentname` with the median `psscore`

   (c) if `type = 2`, return the best `studentname` with the median `(onlinequizscore+psscore)`

Using your best knowledge from CS2010, advise Steven how to implement his leaderboard system.

If you need more space, you can continue your answer for Section 3.1 here

3.2 Graph Data Structure (15 marks)

What is the best graph data structure to store graphs with characteristics like the example shown in Figure 4 below: Near complete graphs of V vertices ($5 < V < 100000$) with only up to k edges missing ($0 \leq k \leq 7$)?

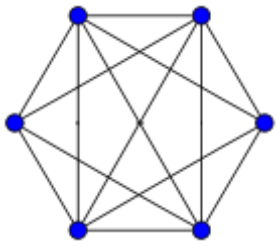

Figure 4: Near Complete Graph with $V = 6$ Vertices (One Edge is Missing)

– End of this Paper –

Candidates, please do not touch this table!

Section	Maximum Marks	Your Marks	Comments from Grader
1	15		
2	50		
3	35		
Total	100		