Assignment 2 (15% of total marks)

Due date: 10 August 2019, Saturday

Scope:

The tasks of this assignment cover the **data structure and algorithm**. The assignment covers the topics discussed in topics 4, and 5.

The assignment is divided into two parts – Part One covers the theoretical aspect of the materials discussed during classes, and Part Two covers the practicality of the concepts. The total marks for Part One is 30, and Part Two is 70.

Assessment criteria:

Marks will be awarded for:

- Correct,
- Comprehensive, and
- Appropriate

application of the materials covered in this subject.

Marks:

Total mark: 100
Weightage: 15% of total subject mark

Assignment Specification:

Part A: (50 marks)

Question 1 (20.0 marks)

a) One main difference between a binary search tree (BST) and an AVL (Adelson-Velski and Landis) tree is that an AVL tree has a balance condition, that is, for every node in the AVL tree, the height of the left and right subtrees differ by at most 1. Starting with an empty BST and AVL tree, insert elements into the two trees with the following keys:
(i) 6, 9, 10, 22, 24 **(5 marks)**
(ii) 6, 14, 7, 10, 12, 13, 15, 9, 11, 6 **(7 marks)**

b) Based on what you have done for Question 3a(i), what is the big-O (**worst case**) complexity of the total time required to build a binary search tree (BST) consisting of n nodes? Explain your answer. Answer without explanation gains no mark. (Hint. The question asks for **worst case** complexity.) **(4 marks)**

c) Similarly, from what you have done for Question 3a(ii), what is the big-o (**worst case**) complexity of the total time required to build an AVL tree consisting of n vertices? Explain your answer. Answer without explanation gains no mark. (Hint. The question asks for **worst case** complexity.) **(4 marks)**

Question 2 (20.0 marks)

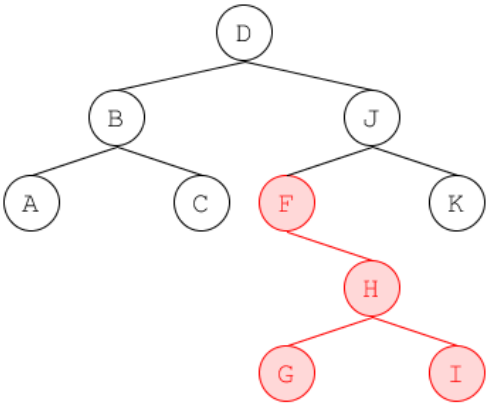
a) Starting with the given 2-4 Tree below, insert elements into the tree with the following keys, **showing the insertion at each step and each split operation performed**: 14, 12, 11, 13, 16, 15. **(10 marks)**

b) Show that a 2-4 tree storing n keys has height $O(\log_2 n)$ **(10 marks)**

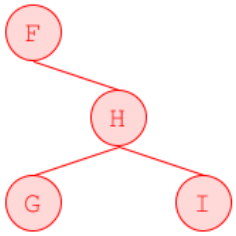
Question 3 (10.0 marks)

a. Describe an algorithm that decides, given any two binary trees *T* and *P*, whether *T* equals a subtree of *P*. For example, the top two diagrams illustrate the scenario where Tree *T* is a subtree of the binary tree *P*; the bottom two diagrams illustrate the scenario where Tree *T* is NOT a subtree of the binary tree *P*. **(7 marks)**

b. Analyse the running time complexity of the algorithm. **(3 marks)**

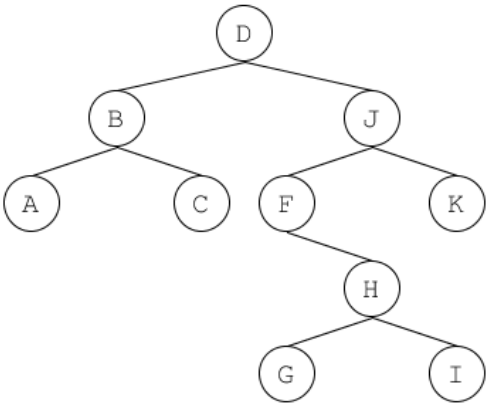


Tree P rooted at D

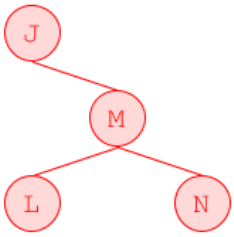


Tree T rooted at F

The tree T rooted at F is a subtree of the binary tree P rooted at D.



Tree P rooted at D



Tree T rooted at J

The tree T rooted at J is NOT a subtree of the binary tree P rooted at D.

Part B: (50.0 marks)

Your task for this assignment is to investigate some of the properties of queues.

You should write a Java or C++ program which simulates the queuing and service of a set of requests at a fast food restaurant.

Input consists of the following data:
- The number of primary servers in the system.
- The number of secondary servers in the system.
- A set of service requests each consisting of an arrival time and two service times. This set is terminated by a dummy record with arrival time and service times all equal to 0. (Note: the arrival times are sorted in ascending order).

For example, the data file

```
3 2
1 2 3
3 3 5
3 2 2
4 3 2
5 2 4
0 0 0
```

Indicates there are 3 primary servers and 2 secondary servers. The first service (customer) comes in in minute 1 (first minute of simulation), and the service requires 2 minutes of primary server's time and 3 minutes of secondary server's time. The second service (customer) comes in in minute 3, and it requires 3 minutes of primary server's time and 5 minutes of secondary server's time, etc.
The last entry of the data file 0 0 0 indicate the end of simulation. (Note that it is possible to have two customers come in the same time as shown in the above sample data (second and third customers).)

Your program should read the name of the data file from standard input and then read the data in the named file into the simulation. For example, the following command will trigger the execution of your program by reading the data file provided:

```
./QueueSim datafile.dat  or

java QueueSim datafile.dat
```

The simulation is to be of a system with two sets of servers, primary and secondary, with a single queue associated with each set. Customers arrive in the system and are served first by a primary server and, on completion of this service, by a secondary server. If all servers of a particular type are busy, the customer will enter either the primary or secondary queue as appropriate.

The simulation should be run until the last customer has left the system.

Output, to standard output, for each version of the queuing process will consist of the following data:

- Number of people served.
- Time last service request is completed.
- Average total service time.
- Average total time in queue(s). Both overall and separate.
- Average length of queue. For each queue and overall.
- Maximum Length of queue. For each queue and overall.
- Total idle time for each server.

Other requirements:

- Software (programming language):
  - Java Version - JDK 6 update 17 or above (Using Windows)
  - C++ / C compiler - g++ 4.0 or above (Using Linux)
- Operating System:
  - Windows XP Professional,
  - Windows Vista Home / Business,
  - Windows 7,
  - Windows 10,
  - Ubuntu Linux 8.04 LTS or above.
- If you use a different environment, please make sure that you MUST check with your lecturers first!
- For C++ solution, students are to give batch / make files for compilation.
- Students are to place all compilation and instructions on how to run the program inside a **readme.txt** file. The markers will refer to this file when marking.
- Programs should be appropriately documented with comments.

Submissions

This assignment is due by **2359 hours Singapore time on Saturday, 10 August 2019**.

- For Part A, type your answer for each question in a MS Word or equivalent document format and save it in a pdf formatted file, name your file as YourName-A2-SolPartA.pdf
- For Part B, the name of your program should be QueueSim.cpp or QueueSim.java, depending on the programming language that you use to develop your program. Execute your program and **screen capture** your output. Next, zip your source code, libraries, readme.txt together with your screen capture and name your file as YourName-A2-SolPartB.zip.
- Zip together YourName-A2-SolPartA.pdf and YourName-A2-SolPartB.zip and name your file as YourName-A2.zip. Do not use your own filename.
- All assignments that do not satisfy the submission requirements listed above will not be evaluated and will be returned to the students with 0 marks.

Submit the files **YourName-A2.zip** through Moodle in the following way:

1) Access Moodle at **http://moodle.uowplatform.edu.au/**
2) To login use a Login link located in the right upper corner the Web page or in the middle of the bottom of the Web page
3) When successfully logged in, select a site **CSCI203 (SP319) Algorithms and Data Structures**
4) Scroll down to a section Submissions of Assignments
5) Click at Submit your Assignment 2 here link.
6) Click at a button Add Submission
7) Move a file, for example, **YourName-A2.zip** into an area. You can drag and drop files here to add them. You can also use a link *Add…*
8) Click at a button Save changes,
9) Click at a button Submit assignment,
10) Click at the checkbox with a text attached: By checking this box, I confirm that this submission is my own work, … in order to confirm authorship of your submission,
11) Click at a button Continue.

A policy regarding late submissions is included in the subject outline.

Only one submission per student is accepted.

Assignment 2 is an individual assignment and it is expected that all its tasks will be solved individually without any cooperation with the other students. Plagiarism is treated seriously. Students involved will likely receive zero. If you have any doubts, questions, etc. please consult your lecturer or tutor during lab classes or over e-mail.

********** End of Assignment Specification **********