Assignment 2 (20% of total marks)

Due date: 12 August 2021, Thursday

Scope:

The tasks of this assignment cover the **data structure and algorithm**. The assignment covers the topics discussed in topics 3 and 4.

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 40, and Part Two is 60.

Assessment criteria:

Marks will be awarded for:

- Correct,
- Comprehensive, and
- Appropriate

application of the materials covered in this subject.

Marks:

Total mark: 100
Weightage: 20% of total subject mark

Assignment Specification:

Part A: (40 marks)

Question 1 (20 marks)

a. The INORDER traversal output of a binary tree is W,O,L,L,O,N,G,O,N,G and the PREORDER traversal output of the same tree is G,L,O,W,O,L,N,N,G,O. Construct the tree and determine the output of the POSTORDER traversal output. **(5.0 marks)**

b. 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) 3, 6, 7, 19, 21

(ii) 10, 6, 2, 5, 15, 12, 3, 8, 11, 7 **(10.0 marks)**

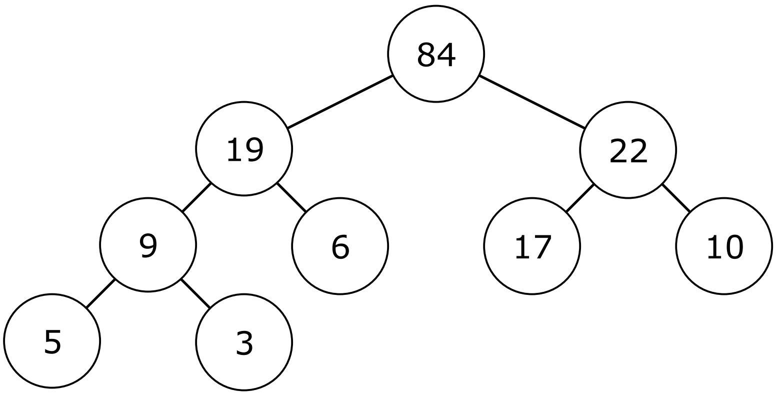
c. Based on what you have done for Question 1b, what is the big-O (**worst case**) complexity of the total time required to **build** a binary search tree (BST) consisting of n nodes? Explain your answer. Answer without explanation gains no mark. (Hint. A tricky question. The question asks for **worst case** complexity. Think properly.) **(5.0 marks)**

Question 2 (20 marks)

a. The operation $convertToMinHeap(h)$ converts a maximum heap into a minimum heap. Design an algorithm of $convertToMinHeap$ that runs in $O(n \lg n)$ time. Show that your algorithm runs in $O(n \lg n)$.

(10.0 marks)

b. Given the following maximum heap, illustrate the process of converting it into a minimum heap using your algorithm described in (a). You need to show the intermediate processes. **(10.0 marks)**



Part B: (60.0 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) arrives 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) arrives 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 arrive 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.

NOTE: Since the question is to assess your understanding of the concept of Queue, you are NOT allowed to use the library of the language that implement queue. You need to write the codes (implementation) of Queue for this exercise. (See point (iii).)

Sample output:

```
YCT-MacBook:csci203 sionggojapit$ ./a2 a2data6.txt
Simulation Start Time: Sat Jul 24 02:41:39 2021

a2data6.txt is used for this simulation.
Start simulation using filename -> a2data6.txt
This simulation uses 6 primary server(s), and 6 secondary server(s).

Total minute of simulation: 277
Number of people served: 100

Total service time: 1888
Average total service time: 18.88 minutes

Average total time in primary server queue: 1 minutes
Average total time in secondary server queue: 0.966667 minutes
Average total time in both queue: 0.983333 minutes

Total primary queue size: 3
Total secondary queue size: 60

Average length of primary queue: 0.0108303
Average length of secondary queue: 0.216606
Average overall length of queue: 0.113718

Maximum length of primary queue: 1
Maximum length of secondary queue: 3
Average maximum length of queue: 2
Maximum length of overall queue: 4

Total idle time for each server:
Primary server 0: 128 minutes
Primary server 1: 98 minutes
Primary server 2: 98 minutes
Primary server 3: 107 minutes
Primary server 4: 115 minutes
Primary server 5: 96 minutes

Secondary server 0: 146 minutes
Secondary server 1: 143 minutes
Secondary server 2: 123 minutes
Secondary server 3: 110 minutes
Secondary server 4: 137 minutes
Secondary server 5: 133 minutes

Simulation ends after 277 minute(s) or 4.61667 hour(s) of simulation process.
Simulation End Time: Sat Jul 24 02:18:51 2021

YCT-MacBook:csci203 sionggojapit$
```

Standard Requirements for Part B (Programming question):

(i) Java Version – JDK 6 update 17 or higher (Using Windows), or
(ii) C++ / C compiler – g++ 4.0 or higher (Using Windows or UBUNTU). In the event that you use UBUNTU via VM, be careful with the memory function of the language and make sure that the functions are used properly and do not cause any segmentation error when the codes are compiled in Windows environment.
(iii) All coding must be your own work. Standard libraries of data structures and algorithms such as STL may not be used.
(iv) Programs should be appropriately documented with comments.
(v) Execute your program and screen-capture the output. Include in your submission all source code and libraries plus the screen-captures.
(vi) Students are to place all compilation and instructions on how to run the program inside a readme.txt file. Your lecturer will refer to this file when marking. Without a readme.txt or clear instructions for compilation, your lecturer will compile based on his/her computer setting; any incompatibility, will deem as failure to compile the program.
(vii) Submission filenames are to follow the naming convention given in the submission instruction below. Do not use your own filename.

Submissions

This assignment is due by 9:00 pm Singapore time on Thursday, 12 August 2021.

- 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 YourUOWStudentNumber-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 YourUOWStudentNumber-A2-SolPartB.zip.
- Zip together YourUOWStudentNumber-A2-SolPartA.pdf and YourUOWStudentNumber-A2-SolPartB.zip and name your file as YourUOWStudentNumber-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 **YourUOWStudentNumber-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 (SP321) 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, **YourUOWStudentNumber-A2.zip** into the submission 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 **********