Project 4 Binary Tree and Heaps

CECS 274 Data Structures Fall 2020

Deadline: 11:55pm, November 22th (Sunday), 2020

Note: the sample codes are given in Python format, you can flexibly choose other languages like Java, etc.

1. (25pt). Implement a Binary Search Tree with inorder traversal. In the tree, you need to design the function to find the minimal and maximal nodes, insert into the tree, delete from the tree, inorder traversal of the tree. Name your code to a file as *1_1_inorder_bst.py.* Note that you may need to add new functions except the defined functions. (You can either use the given sample code or design by yourself)

   Sample outputs:
   Insert the following numbers:
   15, 23, 32, 40, 57, 36, 88
   Output the Min Value: 15
   Output the Max Value: 88
   Inorder traversal of the given tree: 15 23 32 36 40 57 88
   Now delete 40, Inorder traversal of tree: 15 23 32 36 57 88
   Now delete 15, Inorder traversal of tree: 23 32 36 57 88
   Output the new root node: 23

2. (25pt) Design a Priority Queue with the heap implementation. The heap uses maximal heap where the root has the largest value. In this priority queue, you need to complete the functions like inserting a new value, delete the maximal value. Besides, you need to design the swap function. Name your code file to *1_2_priorityQueue.py* (or .java, .etc).

   Sample outputs*:*
   {32, 37, 12, 46, 22, 31}
   {32, 37, 12, 22, 31}
   {32, 66, 42, 12, 22, 56, 31}
   {32, 41, 42, 12, 22, 56, 121, 31}
   {32, 41, 42, 12, 22, 56, 31}

Submission Requirements

You need to strictly follow the instructions listed below:

1) Submit a .zip/.rar file that contains all files.
2) The submission should include your **source code**. Do not submit your binary code.
3) Your code must **be able to compile**; otherwise, you will receive a grade of zero.
4) Your code should not produce anything else other than the required information in the output file.
6) If you code is partially completed, also explain in the begin comment of specific submission what has been completed and the status of the missing parts.
7) Provide **sufficient comments** in your code to help the TA understand your code. This is important for you to get at least partial credit in case your submitted code does not work properly.

Grading criteria:

Details	Points
Submission follows the right formats	10 %
Submitted code has proper comments to show the design details	15 %
Code can be compiled and shows right outputs	75 %

Rubrics for code outputs

	Level 4 (100%)	Level 3 (70%)	Level 2 (40%)	Level 1 (20%)
Code outputs	It is always correct without crashes	It is always correct and eventually it crashes	It is not always correct and eventually it crashes	It is not correct or incomplete