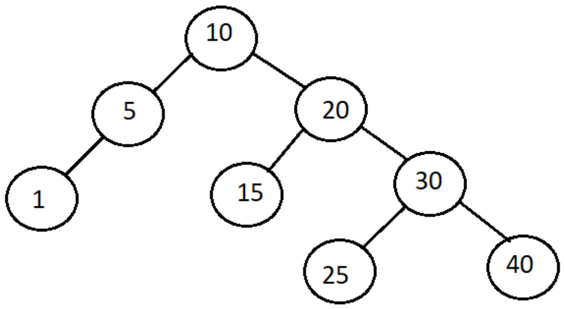
CSDS 233 Assignment 3 Due: Oct 16, 2020, 11:59pm

General instruction: This assignment includes two parts, written and programming. Please write/type your answers neatly so they can be readable. Please submit a single PDF file for the written part and a zip file for the programming part.

Special office hour for this assignment will be on Monday Oct 12, 2020, 11:00am-12:00pm in Zoom (meeting ID: 941 0331 4192, passcode: Binary)

Written Part (50 points):

1. Consider the following tree. (10 points)



i) Is this an AVL tree? Explain why it is or it is not an AVL tree. (2 points)

ii) Insert 21 into the tree. What is the resulting AVL tree? Show your work and draw the tree after insertion. (4 points)

iii) Delete the node with the value of 1 from the result of (ii). Show your work and draw the rebalanced tree after deletion. (4 points)

2. Answer the following questions: (7 points)

i) What is the minimal number of nodes to build an AVL tree of height 6? (2 points)

ii) Determine the truth of the following statements about AVL trees. (2 points)

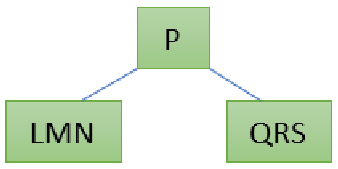
a) Every AVL tree is balanced.
b) Every AVL tree is complete.
c) Some AVL trees are complete.
d) The minimum number of rotations done in an AVL tree insert is 1

iii) Use your own word to briefly explain about balance factor and imbalanced AVL trees. How we can balance an imbalanced tree? You can strengthen your answer by drawing an example of an imbalanced tree and explain your answers using your example (Providing an example is optional). (3 points)

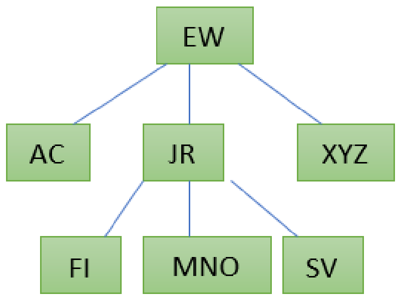
3. Construct a B-Tree of order 4 by using 5, 3, 21, 9, 1, 13, 2, 7, 10, 12, 4, 8. Show all the required steps. (8 points)

4. Determine if the following B-Trees are valid or invalid for branching factor =3. Clearly explain the reasons for your answers. (5 points)

a)



b)



5. Assume we build a binary max-on-top heap from keys 6, 3, 11, 7, 14, 8, 5, 15, 1, 2, 4, 13, 9, 10, 12. (10 points)

   i) Show the result of building this heap by inserting the above keys one at a time in the order given (from left to right), into an initially empty binary heap. Please show key steps and short illustrations if necessary. (6 points)

   ii) Show the heap from problem (i) above after executing three **deleteMaximum** operations on this heap. Show each step. (4 points)

6. Use the same input stream of Q3, execute Heap-Sort in ascending order. Show all the required steps. (10 points)

Programming (50 points):

Create a class named BinarySearchTree with the following methods:

a) `void insert(Node root, int key)` – inserts a node in the binary search tree (BST)
b) `Node search(Node root, int key)` – search a node with a specific key in BST
c) `Node delete(Node root, int key)` – deletes a node from BST
d) `void inorderRec(Node root)` – inorder traversal of BST
e) `Node kthSmallest(Node root, int k)` – find the k'th smallest element in BST

You need to develop this class as efficiently as possible. You are allowed to use helper classes (e.g. Node). Create also a class named Test and in the main method, create an instance of BinarySearchTree. Test methods in BinarySearchTree for this example:

i) Show the results of inserting 2, 1, 4, 5, 9, 3, 6, 7, 10, 12, 11 into an empty Binary search tree
ii) Start with the tree in (i) and delete 4 then delete 9
iii) With the tree from (ii) search 12 and search 4
iv) With the tree from (iii) find the 3rd minimum element in the tree

Please submit a single zip file containing all your *.java files needed for this assignment and a PDF file containing snapshots of outputs obtained. You may also include explanation about added classes and other extra changes you have done into that PDF file.
The submissions will be evaluated on completeness, correctness, and clarity. Please provide sufficient comments in your source code to help the TAs read it.

Grading:

- **BinarySearchTree** implementation: 30 points
- *Test*: 10 points
- Proper encapsulation/information hiding: 5 points
- Design and style: 5 points
  - Style: 2 points. Are variable names descriptive and convey their purpose? Of course, simple concepts like a loop variable do not require descriptive names; e.g., it is perfectly fine, even preferable, to use i for a loop variable. Is formatting clear and consistent?
  - Comments: 1 point. Comments should aid the reader to understand the code. Comments that restate what is already clear from the code are redundant and not helpful. Nor are comments that are not consistent with the code. For each method implementation, state in the comments its worst-case running time.
  - Design: 1 point. Are there lines that never execute? Inelegant constructions? Convoluted or unnecessarily inefficient ways to achieve some result?

Notice: Before your programming assignment is graded, TAs will feed another example to your programs as input (in the Test class), so please make sure your program works for any input before your submission. You can test your code by feeding multiple inputs and check them if they are correct.