Data Structures

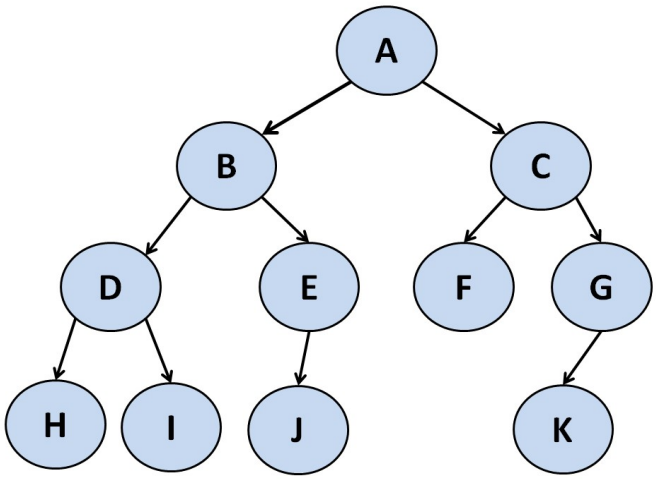
CS202

Assignment 2

Deadline: 6th October 2019

11:55 pm

In case of ambiguity, contact Anusheh Zohair Mustafeez
at 21100072@lums.edu.pk



Q1. This question, you're supposed to make a family tree using the concepts you've learned in class. Modify the tree.cpp file given to implement a family tree with the following functions:

- Constructor
- Destructor
- GetPerson: Given a name of a person, returns a pointer to the node corresponding to that person
- AddChild: Given the name of a person and his/her child, branches a child node from the parent node
- GetParent: Given the name of a person, prints the name of his/her parent
- GetChildren: Given the name of a person, prints the name(s) of his/her children
- GetGeneration: Given the name of a person, returns the generation to which a he/she belongs to
- GetGenerationCount: Given a generation number returns number of people belonging to the same generation
- GetGenerationGap: Given the name of two people, returns the generation gap between them
- Level_traversal: Prints the whole family tree level-wise

Note: Here generation numbers are synonymous to levels in a tree e.g. the root node will be in the first generation and its children will form the second generation and so on.

You are given test1.cpp file to test your code. You are NOT allowed to modify the testing file nor are you allowed to hardcode answers to pass the tests.

Q2. In this part you'll implement a barebone BST. Please refer to the file bst.h, which provides the necessary class definitions and function declarations and modify bst.cpp to implement the following functions for BST:

- Constructor
- Destructor
- insert: inserts the given key-value pair into the tree
- search: takes key k as input and returns pointer to the node that has the matching key. Returns NULL if k does not exist in the tree
- delete_node: deletes node with the given key k
- getRoot: returns root of the tree
- findmin: Given pointer to a node p, finds node with the minimum key in the tree rooted at p
- removemin: Given pointer to a node p, deletes node with the minimum key in the tree rooted at p
- insertHelper: helper function needed by insert for recursion
- remove: helper function needed by delete_node for recursion
- height: Given pointer to a node p, returns the height of the tree rooted at p

Note: Some of the functions in bst.h are relevant to AVL Trees only. You will work with AVL trees in Assignment 3. For now, you can simply comment them out.

You are given test2.cpp file to test your code. You are NOT allowed to modify the testing file nor are you allowed to hardcode answers to pass the tests.

Submission Guidelines:

Submit a zip folder containing the following files on your lms Assignment 2 tab:

- LinkedList.h (unmodified)
- LinkedList.cpp (unmodified)
- queue.h (unmodified)
- queue.cpp (unmodified)
- tree.cpp
- tree.h
- bst.cpp
- bst.h

Naming Convention: A2_*your roll number*.zip