NYU, Tandon School of Engineering
Bridge to Computer Science Program — Summer Extended 2020

Third Midterm Exam

- There are 100 points total.
- Note that there are longer programming problems at the end. Be sure to allow enough time for these.
- We supplied you with a file, named ‘solutions.txt’, where you should type all your answers in.
- For editing this file, you are allowed to use compilers such as Visual Studio, XCODE and CLion
- You may use 2 scratch papers.
- Calculators are not allowed.
- This is a closed-book exam. No additional resourced are allowed.
- Pay special attention to the style of your code. Indent your code correctly, choose meaningful names for your variables, define constants where needed, choose most suitable control statements, etc.
- In all questions you may assume that the users enter inputs as they are asked. For example, if the program expects a positive integer, you may assume that users will enter positive integers.
- No need to document your code in this exam, but you may add comments if you think they are needed for clarity.
- Read every question completely before answering it.

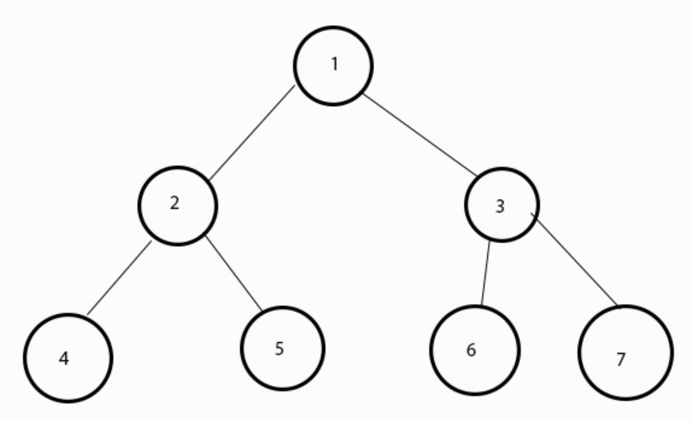
1. (3 pts)An unbalanced binary search tree has a worst-case search time (finding an element in the tree) of.
    a. Theta(N)
    b. Theta (log (N))
    c. Theta(N* log(N))
    d. Theta(N^2)
2. (4 pts) Below, in the line "cout<< *(one.x) <<endl" can the *(one.x) be replaced with a different line of code? If yes, what code could be put here?:

```
class Thing {
public:
       Thing(int newval=42) : {x = newval; }
       int x;
};

int main() {
       Thing* one;
       cout << *(one.x) << endl;
}
```

3. (4 pts) Given the following definition of a tree node (Parent-multi child form):

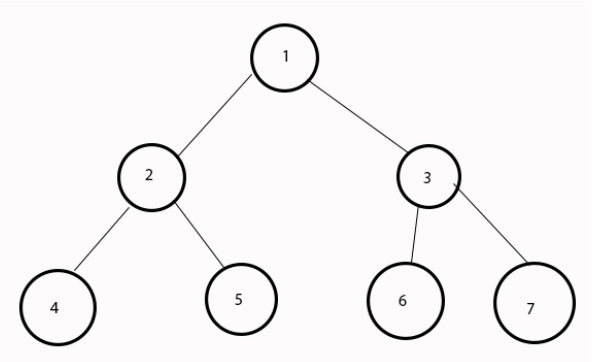
```
class TreeNode {
public:
       int data;
       Vector<TreeNode*> children;
};
```



Provide the code (just a single line) to print (cout) the value in the node labelled 6 above given a pointer (called root) which points to the node labelled 1 above.

4. (5 pts) If a function needs to be defined in the base class, but there isn't code for the function. We can define the function as _________________ which will cause the class to be abstract?

5. (6 pts) Given the tree listed below. Print the in-order traversal.



6. (3 pts) Which of the following will cause the "operator bool()" function on an ifstream object to return true
   a. Successfully opening the file
   b. Failing to open the file
   c. Reaching the end of the file

7. (5 pts) Which of the following data structures might be useful in trying to reverse a set of data stored in a vector in linear time?
   a. A queue
   b. A Stack
   c. A Binary SearchTree
   d. A Map

8. (25 pts) A file on the hard drive contains numerous words separated by spaces. Ask the user for the filename (do not continue until the user provides a name of a file that exists) and then read in all of the words in the file. Record the number of occurrences of each word and output the number of times each word appears. You do not need to sort the results, nor do you need to worry about punctuation or capital/lowercase. Only matching words should be counted. (Performance is not a consideration here, you may use any STL data structures if you'd like)

   For example, if the file is: **the at word the word word at**

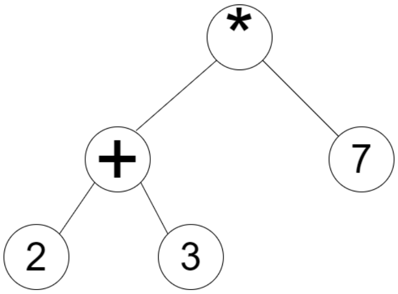
   The result would be: the 2
   at 2
   word 3

9. (10 pts) A number of items have been stored inside of a stack and you would like to output these items in SORTED order (they are unsorted in the stack). Explain, in english, how you would do this, taking into consideration the time complexity of your answer (provide the big Theta runtime of your solution).

10. (10 pts) Assuming a Binary Search Tree node as described below, write a recursive function which will confirm that the tree is "correct." In a correct tree, the parent pointer of any node X will point to a node Y which has X as either a left or right child.

```
class BSTNode {
public:
	int data;
	BST* left;
	BST* right;
	BST* parent;
	//MORE CODE HERE (not provided)
};
```

11. (25 pts) We would like to be able to store an expression tree as a tree of values. An expression tree contains operands and operators and appears as follows:



Each node in an expression tree is either a OperandNode or an OperatorNode. Every node will have a function "calculateValue" which performs the following:

- If the node is an operand, the value is the data (and integer) stored in the class (allow it to be provided to a constructor)
- If the node is an operator (provided to the constructor as a single char), the value is calculated recursively by performing the operation including any child operations.

You are asked to create the classes necessary to store this expression tree.

- You should create the ExpressionTree class and the classes for the nodes.
- You do NOT have to implement any functions other than the calculateValue function. (i.e. you do NOT have to worry about adding nodes or deleting nodes in the tree).
- The only operators we will support are +,-,*,/.
- The ExpressionTree class should have only one function, CalculateValue, which returns the value returned by the call to calculateValue on the root of the tree.