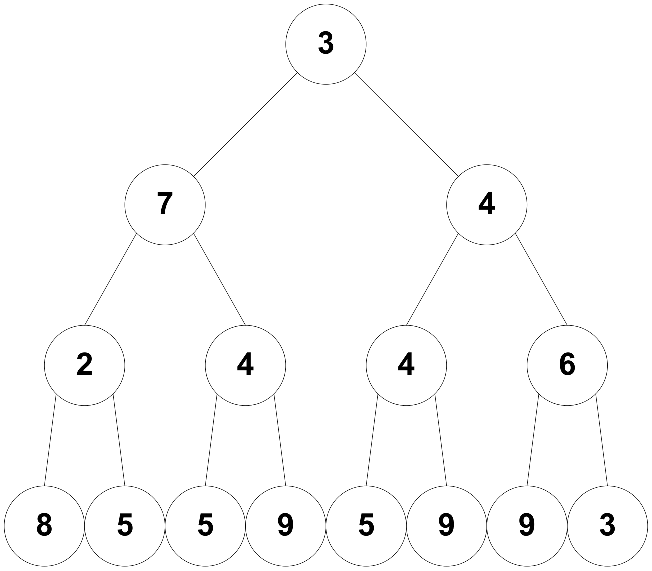
Homework #4

Overview:

You are asked to Create a C++ program that uses Binary Trees to solve both maximum-path and minimum-path finding problems. Your program should read in a series of values and find the maximum path and minimum path from top to bottom as well as displaying the path along the way. As an example, consider a triangle of numbers:

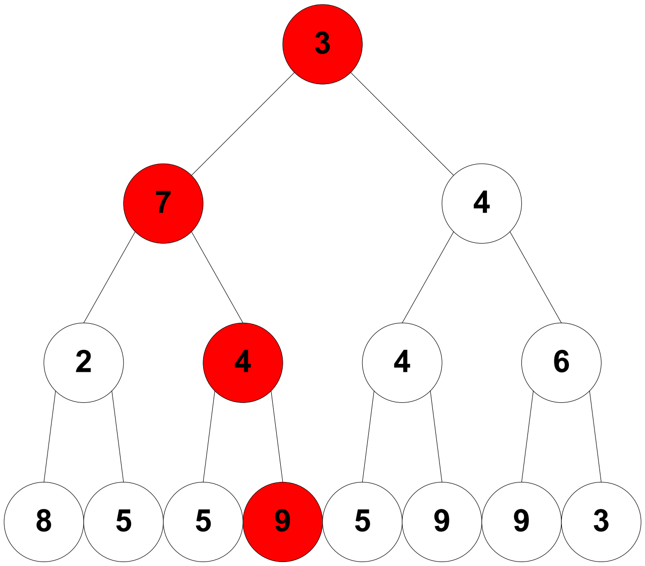
```
   3
  7 4
 2 4 6
8 5 9 3
```

We would like to find a path from the top to the bottom with a maximum total cost and a minimum total. The above triangle can be represented in a binary tree (NOT a BST) as:

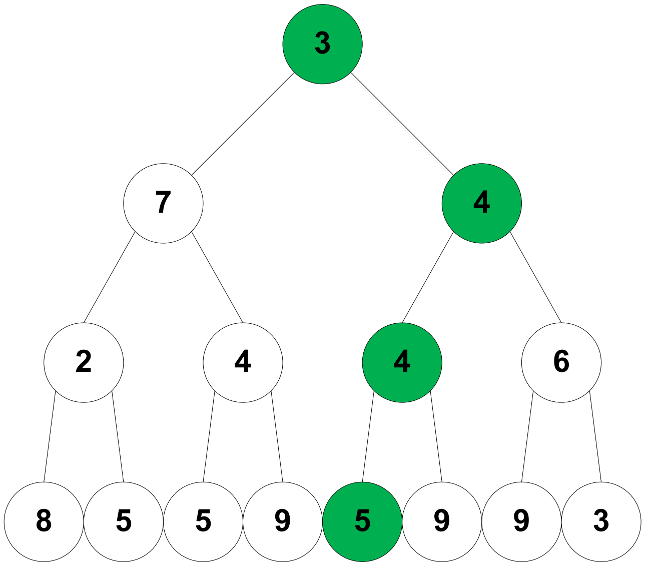


This tree representation reflects the choice we make at each point in the triangle to go to the left or right as we move to the next row. These binary choices mean that the tree will contain repeats of values, but that the choice of movement can be accurately reflected. We can assume that the values will always be non-negative. If two paths have the same total cost, we will prefer the path on the left.

The maximum path through this tree has a total cost of 23. It corresponds to the path:



The minimum path through this tree has a total cost of 16. It corresponds to the path:



Input files
- The input files contain several lines with integer values, separated by spaces as needed. All input files are well formed with valid values. There is no need for exception handling on the inputs and all files follow the same format.
- The first line provides the size of the triangle as the number of rows in the triangle. This value will always be positive and there will always be multiple rows in the triangle. There will never be a degenerate tree that is empty or that contains only a single row.
- The next lines in the file contain the values for each row of the triangle. The first line contains a single value, that will be placed at the root of the tree. Successive lines contain one more value than the previous line. These positive integer values are each separated by a space.

For example, the **input1.txt** contains:

```
4
3
7 4
2 4 6
8 5 9 3
```

This corresponds to a triangle of height 4. The root of this tree has a value 3. It's children (left and right) are 7 and 4. The children of 7 are 2 and 4, the children of 4 and 4 and 6, and so on.

Output files

- Output the maximum path cost through the tree followed by the path (left-most path if there are duplicates.)
- Output the minimum path cost through the tree followed by the path (left-most path if there are duplicates.)

For example, the **ans1.txt** contains:

```
23
3, 7, 4, 9
16
3, 4, 4, 5
```

Note that the order of visiting the cells in the order from the root to the leaves.

Reminder

- Turn in your lab assignment to our Linux server.
- Make sure to only have one (1) .cpp file with the main() function in your working directory, otherwise your program will fail the grading script.
- Ensure that the file names do not contain spaces or the grading script will fail
- Create a folder under your root directory, name the folder hw4 (case sensitive), copy all your .cpp and .h files into this folder, and not into a subdirectory (ArgumentManager.h is also needed)
- Only include the necessary files (.cpp and .h files) in your working directory in your final submission.
- To test your program, you may wish to copy the input files and answer files onto the server and run your program. Do not include any outputs files and after verifying that the code passes the tests, **delete any output*.txt files**.

Please reach out to myself or the TAs for any clarifications or typos.