ITU Computer Engineering Department
BLG 223E Data Structures, Fall 2018
Homework #5
Due 24 December 2018, 23:59

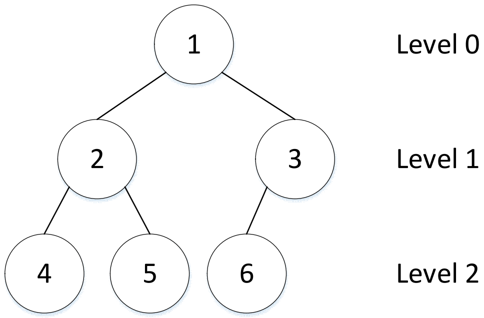
PART I. Construct a Binary Tree

In the first part of the homework, you will construct a binary tree based on the **positive integer** node values given in an input text file. The first line of the input file contains a series of positive integers that will be used on tree nodes as follows. The first integer will be used as the value of the tree root. Then, second integer (if any) will be used as the value of the left child of the root and third integer (if any) will be used as the value of the right child of the root, and so on.

As a general rule, if the value of the parent node is **ith** integer in the file, then the value of the left child of this parent node is **(2*i)th** integer (if any) in the file and the value of the right child of this parent node is **(2*i+1)th** integer (if any) in the file. This method fills in the tree level by level as exemplified below:

The first line of the Input file: 1 2 3 4 5 6

The Tree:

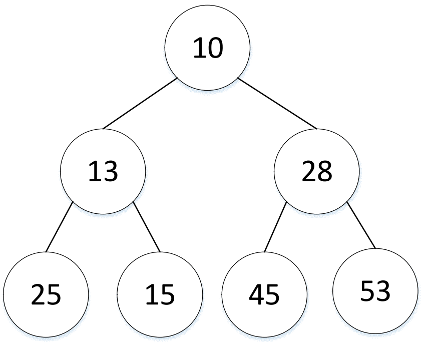


PART II. Path of a Sum in a Binary Tree

In the second part of the homework, you will find all the paths (i.e., **up to two paths**) that starts from the **tree root** and sums up to a **target number** in the constructed binary tree. The target number will be given in the second line of the input file. You have to use **preorder traversal** in your searches and your found paths **must start from the root**. If you find one/two path(s) that sum(s) up to the target, then output “Path Found: “ and the numbers on the path(s). If you can find no paths, then print out “No Path Found”. Please look at the example below:

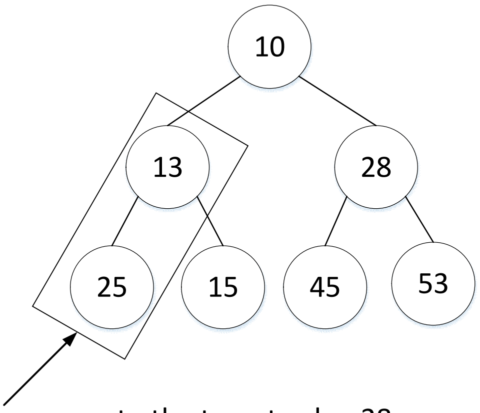
The first line of the Input file: 10 13 28 25 15 45 53
The second line of the Input file: 38

The Tree:



```
The Output:          Path Found: 10 13 15
                     Path Found: 10 28
```

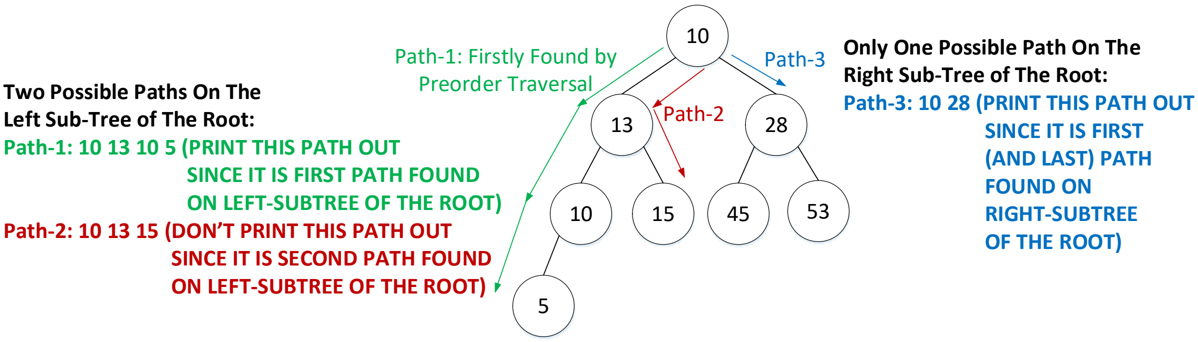
CAUTION: If a subtree contains any path that sums up to the target but **doesn't start from the root** of the tree, then **don't print it out**. See the example below which uses the scenario above and shows a subtree that satisfies the target sum.



A subtree that sums up to the target value 38.
Omit such subtrees, **ONLY print out the trees that start from the root.**

CAUTION: There can be **more than two** paths that sums up to the target and start from the root of the tree. You have to print out **only two of** these paths in that case. You have to print out the first found path on the **left subtree of the root** by using **preorder** traversal. Then, you have to print out the first found path on the **right subtree of the root** by using **preorder** traversal. See the example below which shows three paths that starts from the root and satisfies the target sum. **Please note that, the output only contains two paths.**

The first line of the Input file: 10 13 28 10 15 45 53 5
The second line of the Input file: 38



Two Possible Paths On The Left Sub-Tree of The Root:
Path-1: 10 13 10 5 (PRINT THIS PATH OUT SINCE IT IS FIRST PATH FOUND ON LEFT-SUBTREE OF THE ROOT)
Path-2: 10 13 15 (DON'T PRINT THIS PATH OUT SINCE IT IS SECOND PATH FOUND ON LEFT-SUBTREE OF THE ROOT)

Only One Possible Path On The Right Sub-Tree of The Root:
Path-3: 10 28 (PRINT THIS PATH OUT SINCE IT IS FIRST (AND LAST) PATH FOUND ON RIGHT-SUBTREE OF THE ROOT)

The Output:

```
Path Found: 10 13 10 5
Path Found: 10 28
```

Compilation, Running, and Output

- Your source code file name must be hw5.cpp.
- Your program should compile and run on Linux environment using g++ (version 4.8.5 or later). To compile the code, use the following command:

```
g++ -Wall -Werror hw5.cpp -o hw5
```

- **Your input data filename will be supplied from the command line. Don't name it in your program statically**. Your program must take input filename as command line argument. Then run your executable (named as hw5) using the following command:

```
./hw5 input_file_name
```

where **input_file_name** is name of any test file we will use to evaluate your homeworks. **IF YOUR PROGRAMS DON'T TAKE FILE NAME FROM THE COMMAND LINE, THEY WILL NOT BE EVALUATED AND BE GRADED AS 0.**

- As in the previous homeworks, Your program will be checked using **Calico** (https://bitbucket.org/uyar/calico) automatic checker. Therefore, make sure you **print the messages exactly as given in the homework definition**. You can get zero marks if the automatic checker fails.

- **An example compile, run and output that finds TWO PATHS is as follows:**

  Compile The File: g++ -Wall -Werror hw5.cpp -o hw5
  Input File (e.g., named as input.txt): 10 5 15 20 35 25 30 64 128 256 512
  50
  Run The Program: ./hw5 input.txt
  Program Output:

  ```
  Path Found: 10 5 35
  Path Found: 10 15 25
  ```

- **An example compile, run and output that finds NO PATHS is as follows:**

  Compile The File: g++ -Wall -Werror hw5.cpp -o hw5
  Input File (e.g., no_paths.txt): 10 5 15 20 35 25 30 64 128 256 512
  51
  Run The Program: ./hw5 no_paths.txt
  Program Output:

  ```
  No Path Found
  ```

Submission Rules

- Do not send any e-mails regarding the HW description. If something is not clear to you, **you can ask your question under the thread that is specially started for homework 5 (Questions about HW5) on the message board for BLG 223E on NINOVA.** Please check before writing your question whether your question is asked by someone else. Do not share any code or text that can be submitted as part of your HW.

- Make sure you write your name and number in all of the files of your project, in the following format:

  ```
  /* @Author
  Student Name: <student name>
  Student ID : <student id>
  Date: <date> */
  ```

- Only electronic submissions through Ninova will be accepted.

- You should not share or copy code from your classmates or from the Internet. You should submit your own, individual homework.

- Academic dishonesty, including cheating, plagiarism, and direct copying, is unacceptable.

- Use comments wherever necessary in your code to explain what you did.

- Note that **YOUR CODES WILL BE CHECKED WITH THE PLAGIARISM TOOLS!**