Assignment3
Deadline: 11:59PM Apr 5

Partner Requirement
You can work in a group of up to 3. No exceptions to this group number. Your partner can be from a different set.

Requirements
In this assignment, you are required to write a function that reads integer inputs and add a node and delete a node to a binary search tree. It should be able to perform in-order traversal of the tree and print it to the output.

The delete operation is different from conventional delete operations. If a data is found in the tree, then the node is deleted. If the data is not found, then the node is added.

The input file purely contains integer range values in each line.
```
>> cat input1.txt
1
2
3
4
5
6
8
9
>> cat input2.txt
7
9
>> ./a.out input1.txt intput2.txt output.txt
>> cat output.txt
1
2
3
4
5
6
7
8
```

Below are the specifications of the lab.

1. The run command will be
   ```
   >> ./<name of executable> <input file1> <input file 2> <output
   file>
   ```
2. There will be duplicates.
3. Values in both input files are not sorted.
4. There will be corner case testing in this assignment.

5. No limit on the number of lines in input files
6. Guaranteed to have no empty lines, no integer outside integer range and no data other than int type in the input file
7. Do not add any random trailing new lines or spaces. I will allow up to one new line character at the end of an output file. Example below:

```
>> cat output.txt
1
2
3
4
5 <- having a new line character here is please
>>
```

Below examples will be incorrect

```
>> cat output.txt
1
2
3
4
5>>    <- no new line character
```

Or

```
>> cat output.txt
1
2
3
4
5

    <- extra new line
>>
```

Grading

I will provide sample input and output files. Make sure to test against those to ensure that your program output format works. Any grading failure due to not following specifications will result in 0. For full marks this week, you must:

- (1 point) Correctly submit A number file of you and your partner
- (6 point) Generate a correct solution (including correct memory allocation and deallocation) to the problem(s) in this lab
- (1 point) handle error cases

Extra Bonus

Your job is to make the above operations as fast as possible. I will allow you to change the data structures or operations as long as it lets do the operation as specified. I will not count space complexity into account. I will just run the program and get the time. If you are interested, you can run the following to see how long it takes to run the program.

```
>> time ./<name of executable> <input file1> <input file 2>
<output file>
```

It will report the time it takes to run the program. I will not expose my input, but it will be a rather large input, so it is obvious to see how much performance improvement is achieved. I will rank the speed from the shortest runtime to the longest runtime. Top 30 students will get the bonus. The following is the bonus that will be applied:

- Rank 1-10 students: 5% bonus to the final grade
- Rank 11-20 students: 4% bonus to the final grade
- Rank 21-30 students: 3% bonus to the final grade
- Rank 31-40 students: 2% bonus to the final grade

Unfortunately, the remaining students will not get any bonus point. I will announce the winners by listing all A numbers of the class from the fastest to the lowest. This is an optional part of the assignment, and it will not impact any of your existing grades.

Submission Files
- You must deliver only one .c file named: **assignment3.c** (do not capitalize)
- One or more AXXXX.txt files if you have a partner (empty file, but with your A number as file name. Make sure to include 0's, match this A number with your A number in learning hub, and have .txt extension)
- Github: https://classroom.github.com/a/lL9OKsNR