

CS 211 Data Structures and Algorithms Lab
Spring, 2022-23

Assignment no.	4
Objective	Implement of Binary Search Tree (BST)
Total marks	6
Due (without penalty)	04th June (Sunday) 11:59 pm
Penalty for violating naming convention(s)	10%
Penalty for late submission	10%

The objective of this assignment is to implement ***Binary Search Tree*** (BST).

Command-line argument:

Your program should receive a file (input file) as a command line argument.

Note: Sample input and output files are uploaded with this document.

Input file:

- The input file will be a text file where each line will be of any of the following format:
insert <number>, inorder, preorder, postorder, search <number>, minimum, maximum, successor <number>, predecessor <number>
- Here <number> represents any non-negative integer.
- The input will be given in such a way that, at any point in time, the BST contains only distinct numbers.

Output file:

The output must be in a file named ***output.txt***'. Every line in the input file must have a corresponding output line in output.txt. The details are given below.

Command	Meaning	Output (output.txt file)
insert <number>	Insert <number> to the BST	<number> inserted

inorder	Do an inorder traversal of the BST	Sequence of numbers (separated by a white space) obtained by doing inorder traversal or <empty-line> if BST is empty
preorder	Do a preorder traversal of the BST	Sequence of numbers (separated by a white space) obtained by doing preorder traversal or <empty-line> if BST is empty
postorder	Do a post-order traversal of the BST	Sequence of numbers (separated by a white space) obtained by doing postorder traversal or <empty-line> if BST is empty
search <number>	Search <number> in the BST	<number> found or <number> not found
minimum	Obtain the minimum number in the BST	<minimum-number> or <empty-line> (if BST is empty)
maximum	Obtain the maximum number in the BST	<maximum-number> or <empty-line> if BST is empty
successor <number>	Obtain the successor of <number> in the BST	<successor> / <number> does not exist / successor of <number> does not exist (if <number> is the maximum number) / Tree is empty(if BST is empty)
predecessor <number>	Obtain the predecessor of <number> in the BST	<predecessor> / <number> does not exist / predecessor of <number> does not exist (if <number> is the minimum number) / Tree is empty (if BST is empty)

You can follow your own pseudocode for implementing these functions. **But the ‘effect’ should be the same as that discussed in the class.** For example, we know that a node can potentially be inserted at many places in a BST. ***But for this assignment, it is required that the node should be inserted at the leaf.***

Submission and evaluation:

- Submit a valid .c file. [**.cpp and .txt files as source code are not accepted.**]
- The program you submit should output: **‘output.txt** when we run the program for evaluation, any other names for the output file are not accepted during auto evaluation,

you will be straight away awarded with **0** marks and it is final.

- The main file of your program should be named after your IIT Dharwad roll number. For example, `<roll_no>.c`, where **roll_no** specifies your IIT Dharwad roll number (220010001.c)
 - Do the stress test of your program well before submission.
 - You may use the attached sample input files for testing, the corresponding output files are also attached.
 - We have some hidden inputs with us to test your program. *The marks you obtain are purely based on whether your program correctly gives outputs for the hidden inputs.*
- If your program has only a single source file, please submit the file as it is. If your program has multiple source files, please submit your code as a zip file where the name of the zip file should be your roll number. It is **important that you follow the input/output conventions exactly** (including the naming scheme) as we may be doing an automated evaluation. ***There will be a penalty of 10% (on the mark you deserve) if you do not follow the naming conventions exactly.***
- Follow some coding style uniformly. Provide proper comments in your code.
- *Submit only through moodle. Submit well in advance.* Any hiccups in the moodle at the last minute is never acceptable as an excuse for late submission. Submissions through email or any other means will be ignored.
- Acknowledge the people (other than the instructor and TA) who helped you to solve this assignment. The details of the help you received and the names of the people who helped you (including internet sources, if applicable) should come in the beginning of the main file as a comment. *Copying others' programs and allowing others to copy your program are serious offenses and a deserving penalty(100%) will be imposed if found.*
- To be considered for the evaluation without penalty, you have to submit your program by the due date. ***No single minute relaxation on late submission.***
- **Submissions after 24 hours of the due date will not be accepted.**