

Homework 4 : Summer 2021
Due: Saturday, July 31st by 11:59 pm Online.
100 Pts

Learning Objectives: In the last problem set, we have investigate basic principles and theoretical underpinning of BST and hash tables. Main objective of this assignment is to build on that and get some hands on related to above concepts.

Submission Instruction: see the last page

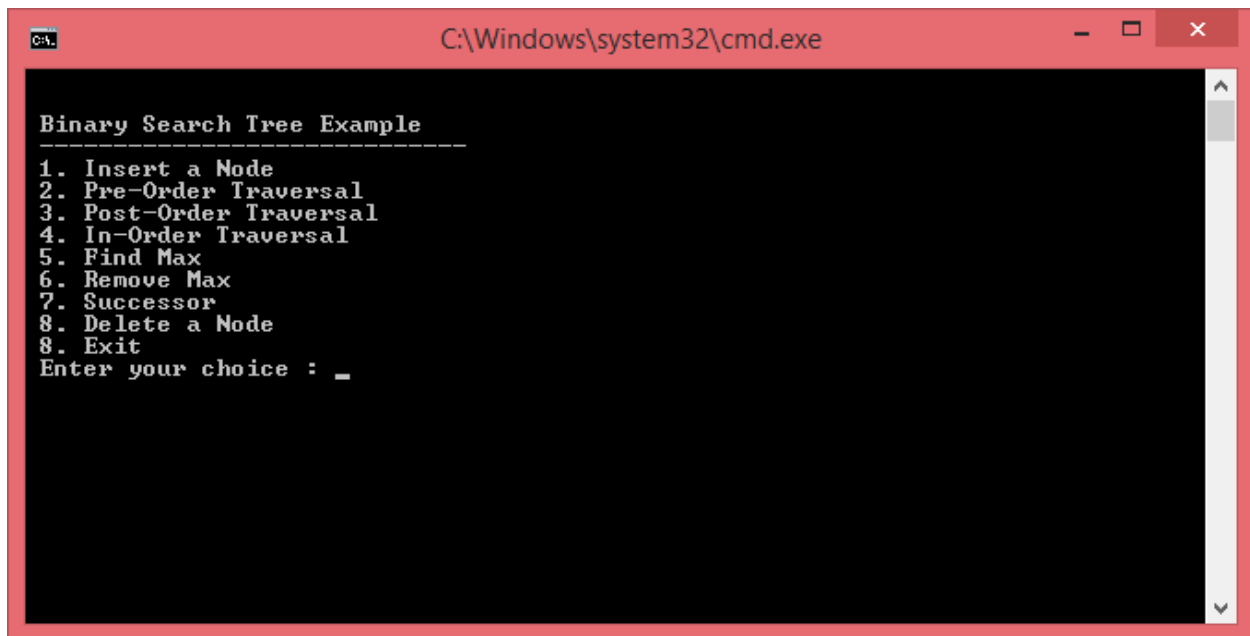
Important: This is an individual assignment. Please do not collaborate or copy any part of the code from other resources such as internet. Any such violations will be considered as an academic interiority policy violations of the class

1.[40 Pts and] Implementation of BST:

Partially completed C++ implementation of BST is given. It implements the INSERT and INORDER traversal. You need to implement the following BST algorithms discussed in the book. Each algorithm should be implemented as a separate method. You are free to change the tree node structure given while implementing these algorithms.

- a) Insert
- b) Post Order Traversal
- c) Pre Order Traversal
- d) Find Max
- e) Remove Max
- f) Successor (see slides for the algorithm)
- g) Delete

Program should have a similar interface given below and the user can select the suitable option to perform the desired operation.



```
C:\Windows\system32\cmd.exe

Binary Search Tree Example
-----
1. Insert a Node
2. Pre-Order Traversal
3. Post-Order Traversal
4. In-Order Traversal
5. Find Max
6. Remove Max
7. Successor
8. Delete a Node
8. Exit
Enter your choice : _
```

2. [60 Pts] Applications of BST

Consider the plane landing problem we have discussed in the class (**This will be discussed on Monday July 26th**) where an airport has one runway and planes request landing times. However, runway access request will be granted only if there is a K time gap from either side of the landing request.

For example, suppose $K = 3$ and a plane requests landing time 56. If there are landing requested for 60 and 45 already granted, then the differences from either sides are $60 - 56 = 4$ and $56 - 45 = 11$. Satisfies the $K=3$ constraint. Landing request will be granted. However, if landing times 58 and 45 have already been granted, then $58 - 56 = 2$, which is less than 3. So, landing request will not be granted.

In this problem, you will be implementing several functionalities of this plane landing system using BST ("Landing times BST"). Your plane landing system should first ask for the K value from the user, then implement the following functionalities.

Your landing times BST should store the landing time (use as the key) and the plane flight number in the node.

- A) Request landing – input for this functionality is the landing time. You can consider the current time as zero. If the landing time request satisfies the K constraint, grant the landing time and insert the landing time to the landing times BST.
- B) Withdraw landing request – A plane can withdraw landing request. This may be due to delay and cancel flight. In this case, remove the landing request from the landing BST.
- C) List landing times and the flight number – display the landing times and plane information in the landing times BST.

Write a C or C++ program that Implements above functionality. You need to use pointers similar to the implementation in problem 1.

SUBMISSION INSTRUCTIONS

- A) Complete code that implement the BST and its functionality for question 1
- B) Complete code that implement question 2

Make a zip file called hw4.zip and submit online