ITU Computer Engineering Department
BLG 223E Data Structures, Spring 2022
Recitation #5

## Problem Definition

In this recitation, given **Complete Binary Search Tree** is converted to **Min Heap** recursively. Firstly user, enters the int value of the nodes of Binary Search Tree. After that, create a Binary Search Tree using these nodes and this tree will be made complete tree. Complete Binary Search Tree Construction has been provided by the program.

After complete binary search tree construction, you should convert this tree to min heap completing below functions in main.cpp file. It is also wanted that user can print all nodes less than a value x in this Min Heap. You should do followig transformation in these functions.

- void inorder_traversal(Node* root, vector¡int¿& arr):
  - This function takes two parameters. Root of the Binary Search Tree and a vector. Firstly vector is empty. Using inorder traversal, you should add the value of the nodes of Binary Search Tree to vector recursively. Thus, the nodes added to the vector will be in ascending order.
- void BST_to_MinHeap(Node* root, vector¡int¿ arr, int* i)
  - This function takes three parameters. First parameter is the root of the Complete Binary Search Tree. Second parameter is a vector which holds the values of the nodes of the Binary Search Tree in ascending order(these values are created in inorder_traversal function). Third parameter is used for index of the vector.
  - Using preorder traversing, you change the value of the Complete Binary Search Tree Nodes with vector values one by one recursively. How the function works is shown in the figures below.

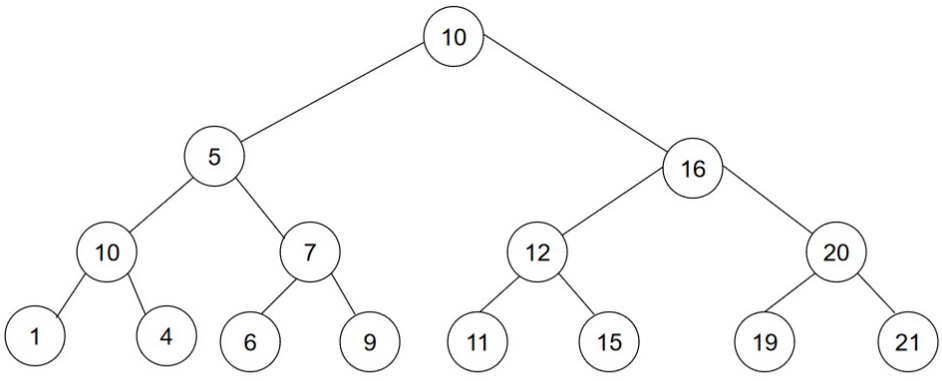


Figure 1: Before BST_to_MinHeap function calls, after inorder_traversal function; Binary Search Tree and vector should be as in the figure.

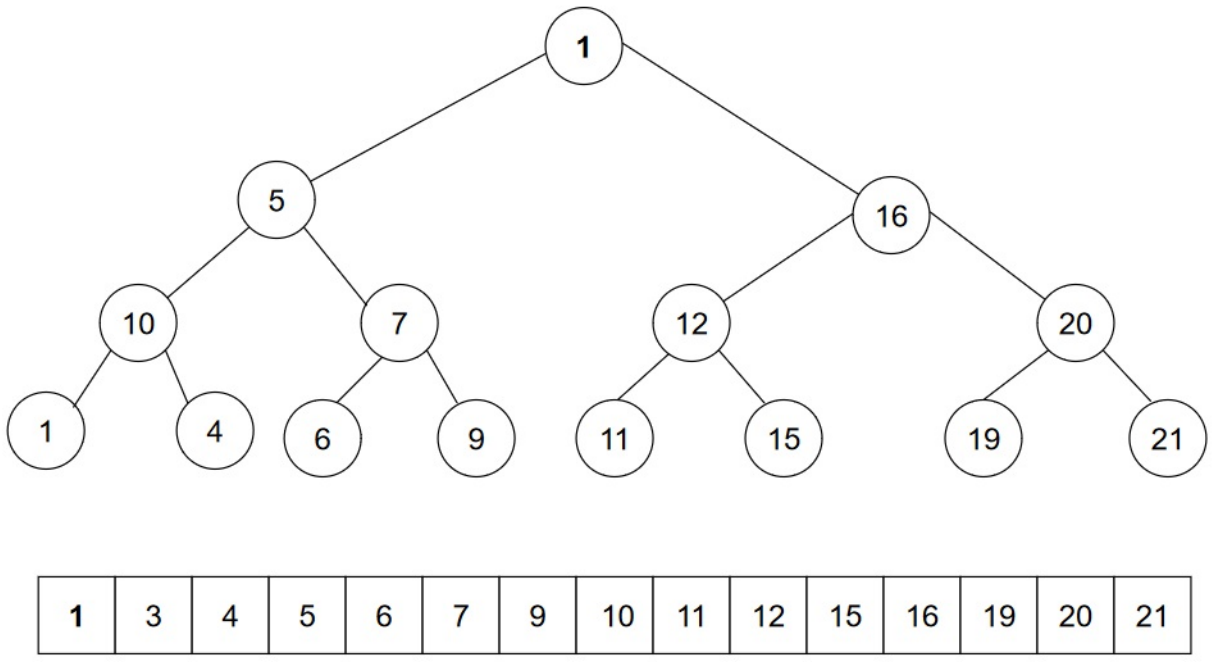


Figure 2: Preorder traversing is done in BST_to_MinHeap function, so firstly value of the root of the Binary search tree is changed with the first element of the vector as in the figure.

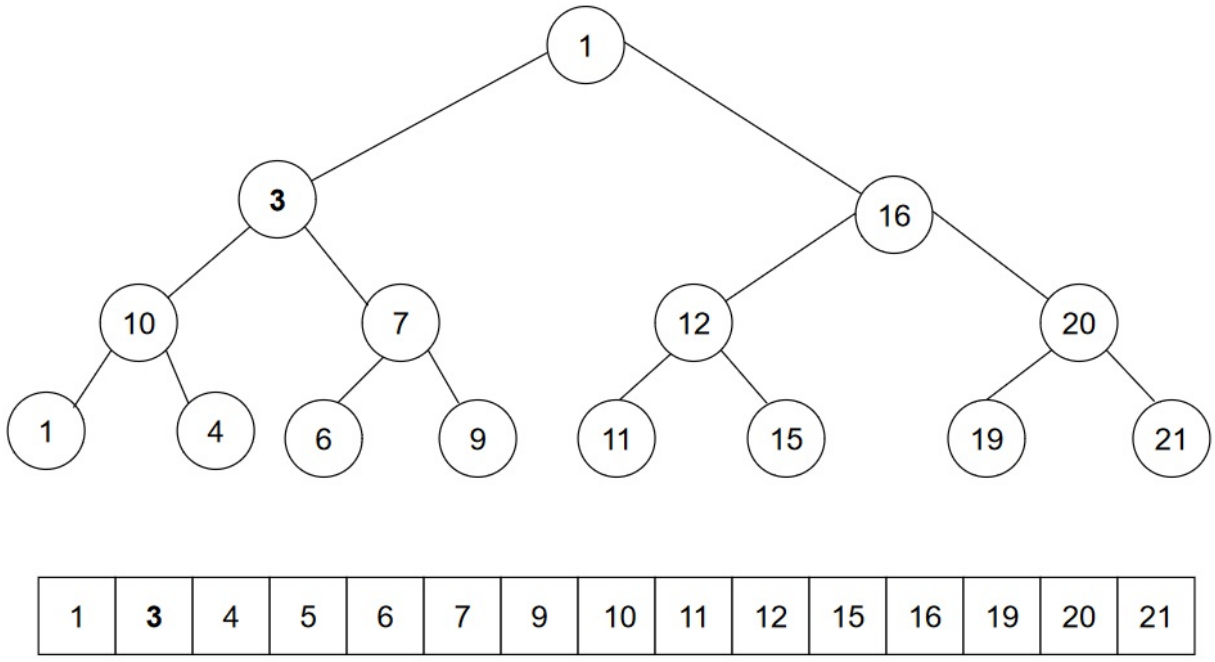


Figure 3: Preorder traversing is done in BST_to_MinHeap function, so secondly value of the first left child of the Binary search tree is changed with the second element of the vector as in the figure.

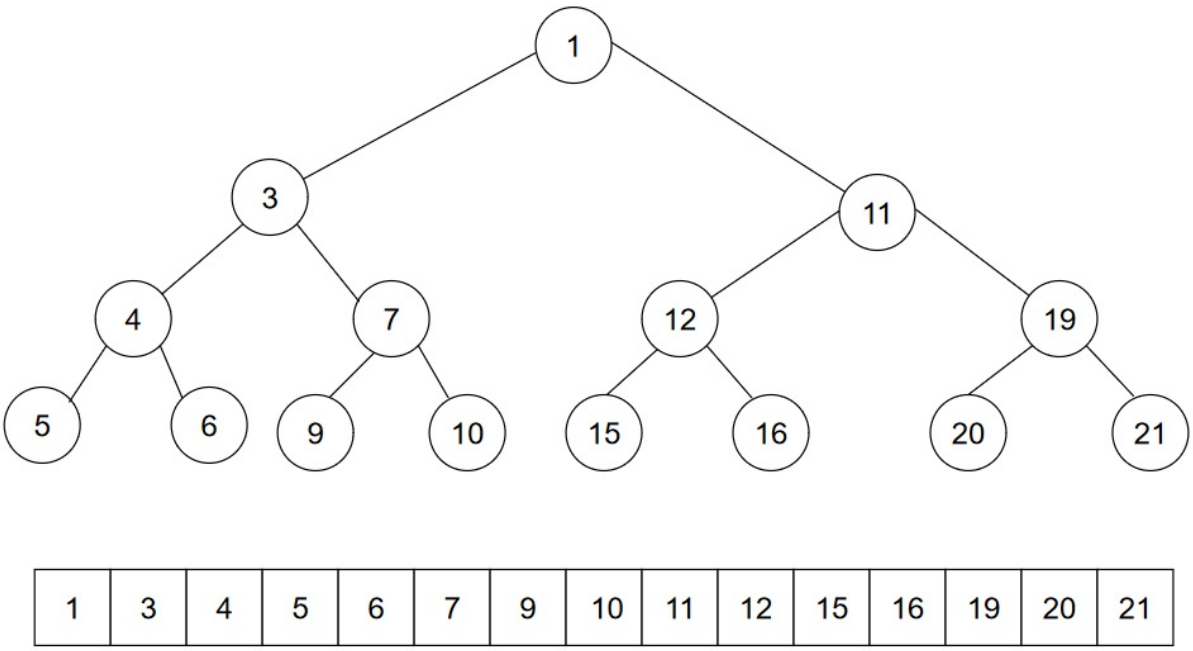


Figure 4: Finally, preorder traversing is completed in BST_to_MinHeap function. Binary Search Tree should be as in the figure. It is a Minheap.

- void print_less_than(int n, Node* root)
  - This function takes two parameters. First parameter is n which is the value of user wants to print all nodes less than n. Second parameter is the root.
  - Using preorder traversing and recursive, you should write the nodes value of the Min Heap(converted from Complete Binary Search Tree) which is less than x.

# 1 Example

Terminal Screen

```
Choose an operation
A: Add node to Binary Search Tree
C: Convert Binary Search Tree to Min Heap
L: List Heap node less than the entered value
P: Print Min Heap
E: Exit
Enter a choice A,C,L,P,E:A
Enter node data:10
Choose an operation
A: Add node to Binary Search Tree
C: Convert Binary Search Tree to Min Heap
L: List Heap node less than the entered value
P: Print Min Heap
E: Exit
Enter a choice A,C,L,P,E:A
Enter node data:5
```

```
Choose an operation
A: Add node to Binary Search Tree
C: Convert Binary Search Tree to Min Heap
L: List Heap node less than the entered value
P: Print Min Heap
E: Exit
Enter a choice A,C,L,P,E:A
Enter node data:15
Choose an operation
A: Add node to Binary Search Tree
C: Convert Binary Search Tree to Min Heap
L: List Heap node less than the entered value
P: Print Min Heap
E: Exit
Enter a choice A,C,L,P,E:A
Enter node data:3
Choose an operation
A: Add node to Binary Search Tree
C: Convert Binary Search Tree to Min Heap
L: List Heap node less than the entered value
P: Print Min Heap
E: Exit
Enter a choice A,C,L,P,E:A
Enter node data:7
Choose an operation
A: Add node to Binary Search Tree
C: Convert Binary Search Tree to Min Heap
L: List Heap node less than the entered value
P: Print Min Heap
E: Exit Enter a choice A,C,L,P,E:A
Enter node data:11
Choose an operation
A: Add node to Binary Search Tree
C: Convert Binary Search Tree to Min Heap
L: List Heap node less than the entered value
P: Print Min Heap
E: Exit
Enter a choice A,C,L,P,E:A
Enter node data:20
Choose an operation
A: Add node to Binary Search Tree
C: Convert Binary Search Tree to Min Heap
L: List Heap node less than the entered value
P: Print Min Heap
E: Exit
Enter a choice A,C,L,P,E:A
Enter node data:1
Choose an operation
A: Add node to Binary Search Tree
C: Convert Binary Search Tree to Min Heap
L: List Heap node less than the entered value
P: Print Min Heap
E: Exit
```

```
Enter a choice A,C,L,P,E:A
Enter node data:4
Choose an operation
A: Add node to Binary Search Tree
C: Convert Binary Search Tree to Min Heap
L: List Heap node less than the entered value
P: Print Min Heap
E: Exit
Enter a choice A,C,L,P,E:A
Enter node data:6
Choose an operation
A: Add node to Binary Search Tree
C: Convert Binary Search Tree to Min Heap
L: List Heap node less than the entered value
P: Print Min Heap
E: Exit
Enter a choice A,C,L,P,E:A
Enter node data:9
Choose an operation
A: Add node to Binary Search Tree
C: Convert Binary Search Tree to Min Heap
L: List Heap node less than the entered value
P: Print Min Heap
E: Exit
Enter a choice A,C,L,P,E:A
Enter node data:12
Choose an operation
A: Add node to Binary Search Tree
C: Convert Binary Search Tree to Min Heap
L: List Heap node less than the entered value
P: Print Min Heap
E: Exit
Enter a choice A,C,L,P,E:A
Enter node data:16
Choose an operation
A: Add node to Binary Search Tree
C: Convert Binary Search Tree to Min Heap
L: List Heap node less than the entered value
P: Print Min Heap
E: Exit
Enter a choice A,C,L,P,E:A
Enter node data:19
Choose an operation
A: Add node to Binary Search Tree
C: Convert Binary Search Tree to Min Heap
L: List Heap node less than the entered value
P: Print Min Heap
E: Exit
Enter a choice A,C,L,P,E:A
Enter node data:21
Choose an operation
A: Add node to Binary Search Tree
C: Convert Binary Search Tree to Min Heap
L: List Heap node less than the entered value
P: Print Min Heap
E: Exit
```

```
Enter a choice A,C,L,P,E:C
Choose an operation
A: Add node to Binary Search Tree
C: Convert Binary Search Tree to Min Heap
L: List Heap node less than the entered value
P: Print Min Heap
E: Exit
Enter a choice A,C,L,P,E:L
Enter value:12
1 3 4 5 6 7 9 10 11
Choose an operation
A: Add node to Binary Search Tree
C: Convert Binary Search Tree to Min Heap
L: List Heap node less than the entered value
P: Print Min Heap
E: Exit
Enter a choice A,C,L,P,E:
```

## Submission Rules

- 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> */

- Use comments wherever necessary in your code to explain what you did.
- Your program will be checked by using **Calico(https://bitbucket.org/uyar/calico)** automatic checker.
- Do not share any code or text that can be submitted as a part of an assignment (discussing ideas is okay).
- Only electronic submissions through Ninova will be accepted no later than deadline.
- You may discuss the problems at an abstract level with your classmates, but 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.
- If you have any question about the recitation, you cand send e-mail to Yunus Emre Cebeci(cebeci16@itu.edu.tr).
- Note that **YOUR CODES WILL BE CHECKED WITH THE PLAGIARISM TOOLS!**