CSE225 Data Structures – FALL 2023

PROJECT #1 (Due December 8, 2023, Friday at 23:59)

In this project, you will implement a special tree structure which satisfy following conditions:

- Each node will have a key value and frequency value (Frequency value represents search frequency of the key value)
- It will be a binary search tree (BST), so
  - each node has at most two children (left child and right child)
  - left child has smaller key value than the node itself
  - right child has greater key value than the node itself
- Every parent node has greater frequency value than all of its children.

You will implement this in two different ways.

a) Offline version:

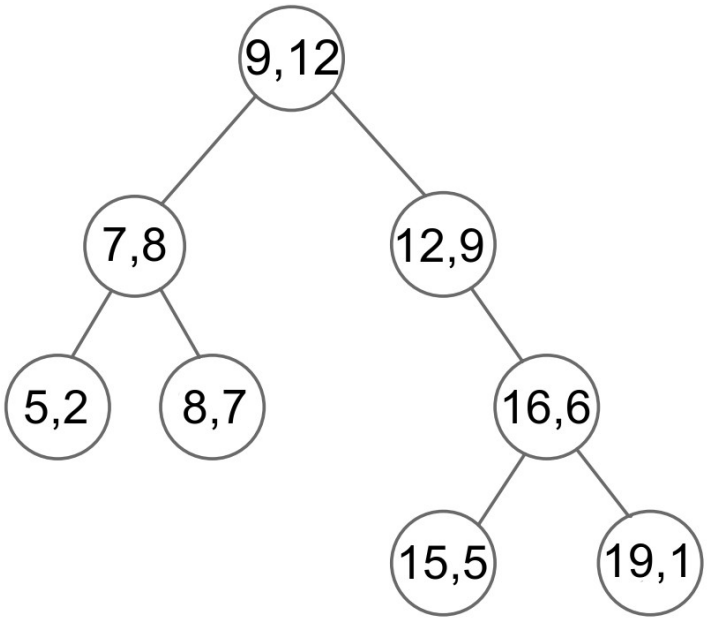
- Your program will construct this tree using a given input file that contains key values and corresponding frequency values.

Example:

Execution

```
>prj1.exe input.txt
>Pre-order traversal of constructed tree : 9, 7, 5, 8, 12, 16, 15, 19
```

input.txt
5, 2 7, 8 8, 7 9, 12 12, 9 15, 5 16, 6 19, 1



Constructed tree

b) Online version:

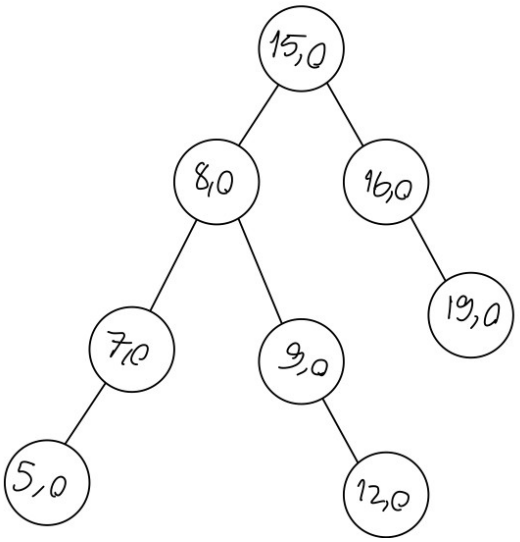
- Your program will construct the tree using a given input file that contains key values (like normal BST). At the beginning frequency values of each nodes will be 0.
- After constructing the tree, it will prompt the user to enter a number to search. When a value is searched, frequency value of corresponding node will be increased by 1.
- The program will update the tree for each search, increasing the frequency of the node and adjusting the node arrangement to satisfy the specified conditions.
- After each search your tree must perform necessary rotations to satisfy the conditions.
- After completing all necessary rotations it will print pre-order traversal of the tree.

Example:

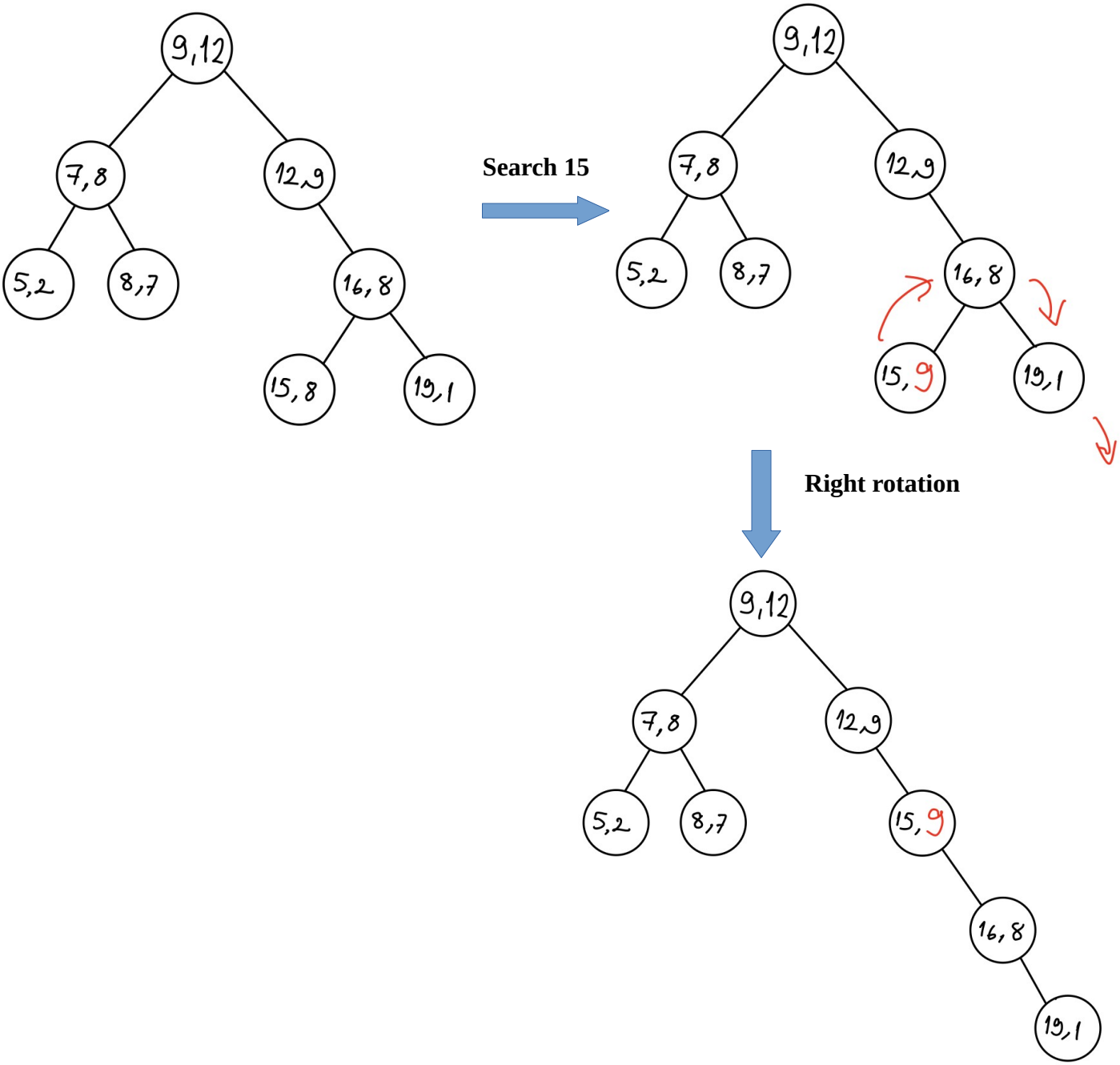
Execution of your program
>myprog.exe input.txt >Pre-order traversal of constructed tree : (15, 0), (8, 0), (7, 0), (5, 0), (9, 0), (12, 0), (16, 0),(19, 0)

input.txt
15 8 7 16 9 5 19 12

Your program will add all the key values in the input file with a search frequency of 0.

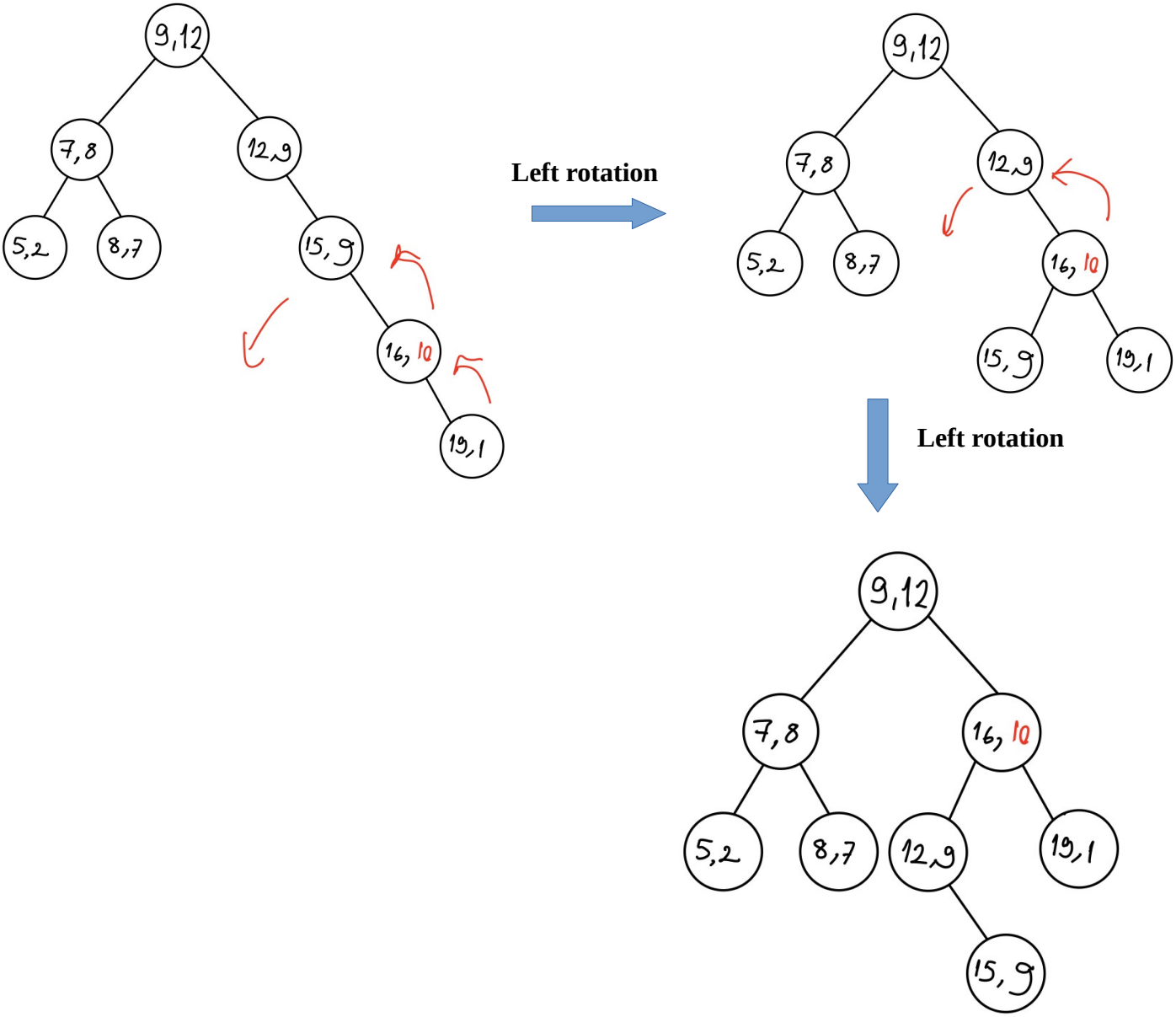


Output of your program (...After some search operations...)
>Pre-order traversal of constructed tree : (9,12), (7,8), (5,2), (8,7), (12,9), (16,8), (15,8), (19,1) >Enter a value to search: **15** >Pre-order traversal of constructed tree : (9,12), (7,8), (5,2), (8,7), (12,9), (15,9), (16,8), (19,1)



Output of your program
>Enter a value to search: **16** >Pre-order traversal of constructed tree : (9,12), (7,8), (5,2), (8,7), (12,9), (15,9), (16,9), (19,1) >Enter a value to search: **16** >Pre-order traversal of constructed tree : (9,12), (7,8), (5,2), (8,7), (16,10), (12,9), (15,9), (19,1)

Search 16 × 2 times



Notes:

- This projects will be done within a group. (2 Students)
- Grading: 90 pts for one part (either part (a) or part (b)), 60 pts for second part. (Total points for the project are 150)
- C programming language will be used.
- Submission of all projects is mandatory to pass the class.
- Students will grade their partners.
- If you cheat (by copying from web, other students, or AI tools) you will get 0.
- Your code will be checked using plagiarism detection tools.
- By the due date, you are required to electronically submit the source code of your program to Canvas. The file name should be in the format StudentID1_StudentsID2_Prj1.c.
- If you have multiple c files, zip them and submit as a single zip file. The file name should be in the format StudentID1_StudentsID2_Prj.zip.