

Dept. of Computer Science and Engineering
Foundations of Algorithm Design and Machine Learning
Assignment 2: Searching

Total marks: 50. Both questions carry equal marks (25 each).

In this assignment, we will explore the following concepts studied in the lecture:

1. Searching
2. Binary Search Tree (BST)
3. Balanced BST

It involves writing an efficient and well commented code for the given Problem Statements. There are two C files with main function and other functions defined already. You are required to edit these files according to the problem and submit both the files. You can add functions if required.

1. Problem Statement:

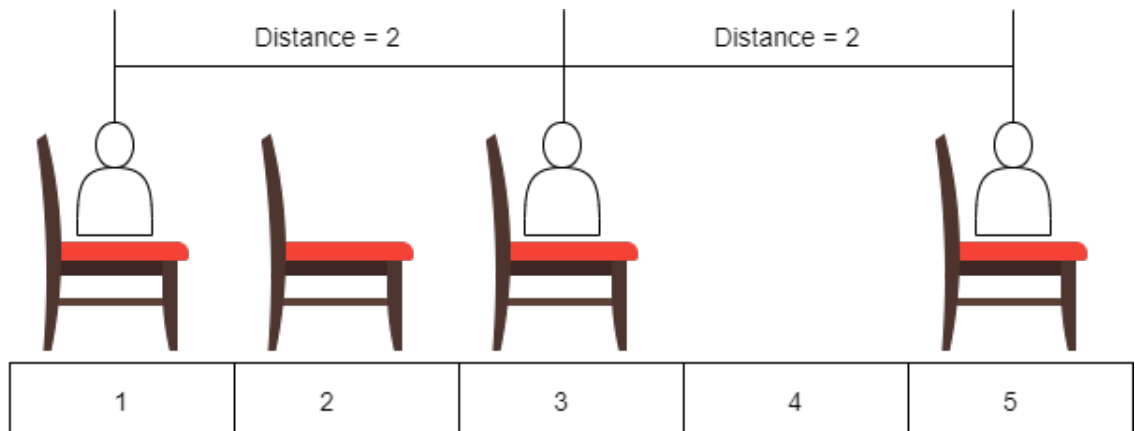
IIT KGP decides to conduct the examination for students after the reopening of college as per Covid-19 guidelines. The students are made to sit in a straight line, so empty chairs are placed in a straight line. There are n empty chairs, the i^{th} chair is at **position[i]** and m is the number of students who need to be seated such that the **distance** between all pairs of students is maximized.

Given the sorted integer array **position** and the integer m . Print *maximum possible distance* between any pair of students.

Constraints:

- $n == \text{position.length}$
- $2 \leq n \leq 10^5$
- $1 \leq \text{position}[i] \leq 10^9$
- All integers in position are distinct.
- $2 \leq m \leq \text{position.length}$

Example 1:



Input: n = 4, position = [1,2,3,5], m = 3

Output: 2

Explanation: Placing the 3 students on chairs 1,3 and 5 will make the maximum distance 2. We cannot further maximize the distance more than 2.

Example 2:

Input: n = 7, position = [1,2,3,4,7,8,11], m = 4

Output: 3

Explanation: Placing the 4 students on chairs 1,4, 7, and 11 will make the maximum distance 3.

Example 3:

Input: n = 6, position = [1,2,3,4,5,1000000000], m = 2

Output: 999999999

Explanation: We can use chairs 1 and 1000000000.

2. Problem Statement:

A professor grades his student in a way such that each student gets unique grades.

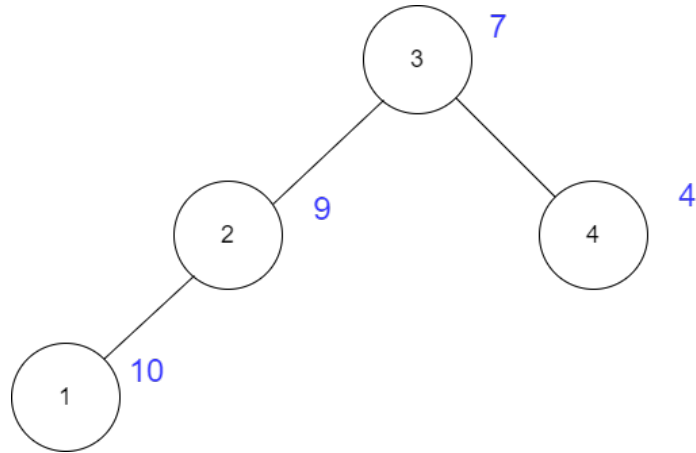
Further, the professor wants to apply a different kind of normalization for grading the students in order to determine the rank of the students. The normalization method adopted by the professor is such that the marks of n students stored as nodes in the Binary Search tree (BST) and each mark is added with the marks of nodes that are greater than the original marks of that particular node, this value will be called the **rank** of the student.

Print the preorder, postorder, and inorder traversal of the ranks of n students according to the original BST.

Constraints:

- The number of nodes in the tree is in the range [1, 100].
- $0 \leq \text{Node.val} \leq 100$
- All the values in the tree are unique.
- It is guaranteed to be a valid binary search tree.

Example 1:



Input: n=4,
 3
 2
 4
 1

Output: *Inorder* : 10,9,7,4
Preorder : 7,9,10,4
Postorder : 10,9,4,7

Example 2:

Input: n=6,
 4
 6
 1
 2
 5
 0

Output: *Inorder* : 18, 18,17,15,11,6
Preorder : 15,18,18,17,6,11
Postorder : 18,17,18,11,6,15