CSE-2212: Design and Analysis of Algorithms-I Lab
Practice Lab 7– June 17, 2025

Experiment: Implementing Greedy Algorithms and divide and conquer algorithms in Java.

Problem Statement 1: Binary Search Tree

Problem Statement: Write a program in Java to implement a Binary Search Tree (BST). Your program should include methods to:

1. insert(num): Add a node to the BST while maintaining the binary search tree property:
    a. All values in the left subtree of a node are less than the node's value.
    b. All values in the right subtree of a node are greater than the node's value.
2. search(key): Check whether a given value exists in the BST. It should return True/False.
3. inorder_traversal(): Print all values in the BST in ascending order using in-order traversal.

The input will have 3 lines

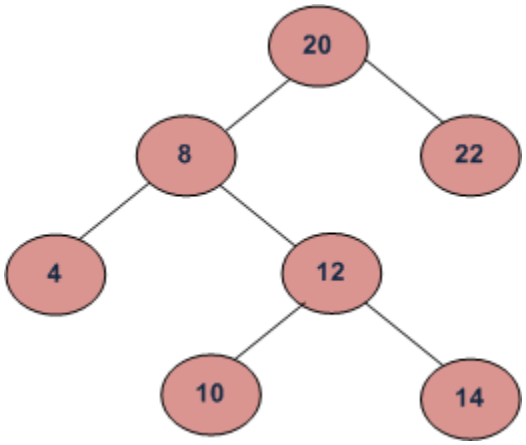
1. an input N - the number of nodes in the BST.
2. N comma-separated integers in a line, each to be inserted into the tree. After all nodes have been inserted, call the inorder_traversal() method.
3. The last line of the input will be the value to search in the three. If the value X is found, it should print "X is found in the BST.". If it is not found, it should print "X is not found in the BST"

Example:
Input:
5
50, 30, 20, 40, 70, 60, 80
40

Output:
40 found in the BST.
In-Order Traversal: 20 30 40 50 60 70 80

Problem Statement 2: Lowest Common Ancestor in Binary Search Tree

In this problem, you have to extend the previous problem's solution. You have to add a new function, **lca(n1, n2)** where Given two values n1 and n2 in a Binary Search Tree, find the Lowest Common Ancestor (LCA). You may assume that both values exist in the tree. The lowest common ancestor between two nodes n1 and n2 is defined as the lowest node that has both n1 and n2 as descendants (where we allow a node to be a descendant of itself). In other words, the LCA of n1 and n2 is the shared ancestor of n1 and n2 that is located farthest from the root [i.e., closest to n1 and n2]. **The time complexity of your implementation should be O(h) where h is the height of the tree.**



Input: lca(10,14)
Output: 12
Explanation: 12 is the closest node to both 10 and 14 which is an ancestor of both the nodes.

Input: lca(8,14)
Output: 8
Explanation: 8 is the closest node to both 8 and 14 which is an ancestor of both the nodes.

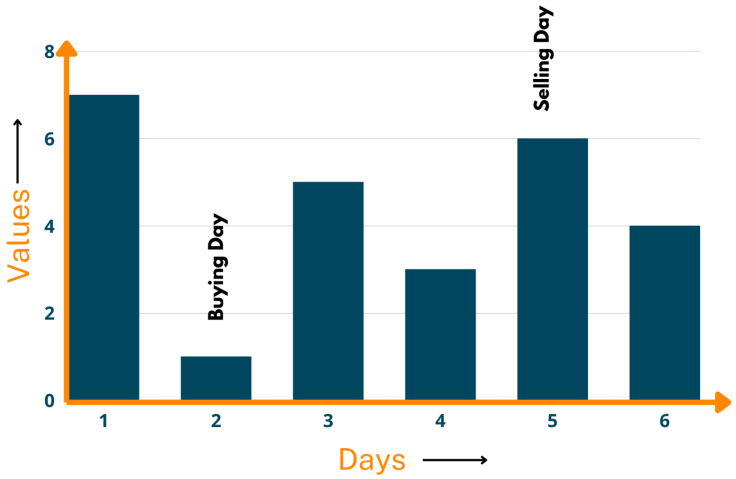
Problem Statement 3: Best Time to Buy and Sell Stock

Problem Statement: You are given comma separated prices which contain the price of a stock on some days. You have to choose one day for buying the stock and a different one for selling it. Return the days on which you buy and sell the stock.

Example:

Input:
6
7,1,5,3,6,4

Output: 5



Problem Statement 4: Minimum Number of Notes

Problem Statement: We are given a value of V and we need to give the change of it in Bangladesh notes. Now the denominations are 1, 2, 5, 10, 20, 50, 100, 200, 500, 1000. The result should contain the least possible change of the given value.

Example:

Input: V=70

Output: 2

Explanation: To get the change of 70, we'll require one note of 50 and another of 20.

Problem Statement 5: Fractional Knapsack Problem

Problem Statement: You are given the weights and values of N items. You need to put these items in a knapsack of capacity W in such a way that the final value of the knapsack is maximum. You can break down the items into fractions to attain the most optimal answer.

1. First line of input will be N
2. The following N lines will each have two integers - V and W for each item
3. Knapsack capacity W

Example:

Input:
3
60 10
100 20
120 30
50

Output: 240

Explanation: Now, the knapsack weight is 50 kgs so you need to add elements with weight equal to this. So, we could have taken the 2nd and 3rd elements but that would exclude the 1st one. Hence, we took 1st and 2nd and then took a fractional (2/3rd) part of the 3rd element. Therefore, getting the maximum possible answer. You can also take a look at the illustration below.