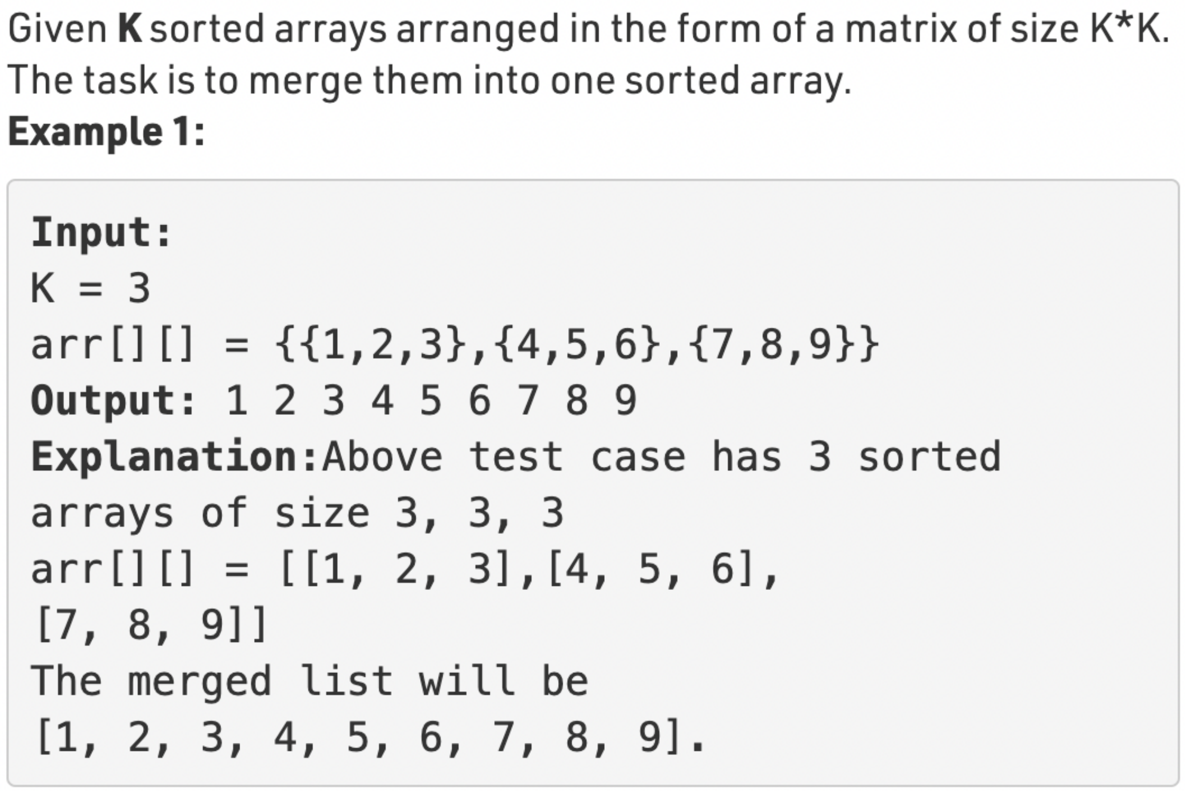
Agenda:

- Merge K sorted arrays (problem statement in class-11-doc)
- Find Path to a given node
- Nodes at a distance K from a target node
- Discuss: Find all nodes at a distance K from any leaf node
- Iterating a BST in a sorted manner without storing all the elements in an array. Imp application:
    - Problem: Check if a pair with a given sum exists in a BST or not

      Solution: Simply create two BST iterators and check!

Merge K sorted arrays

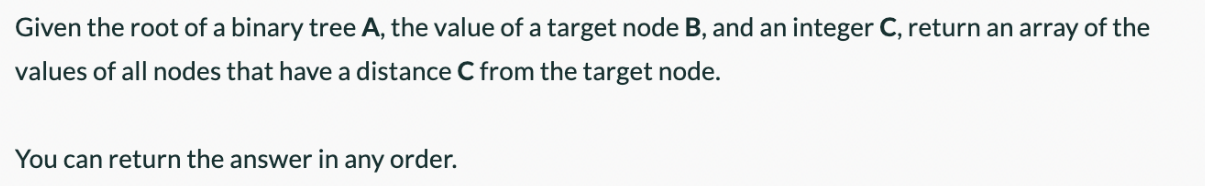


Given **K** sorted arrays arranged in the form of a matrix of size K*K. The task is to merge them into one sorted array.

Example 1:

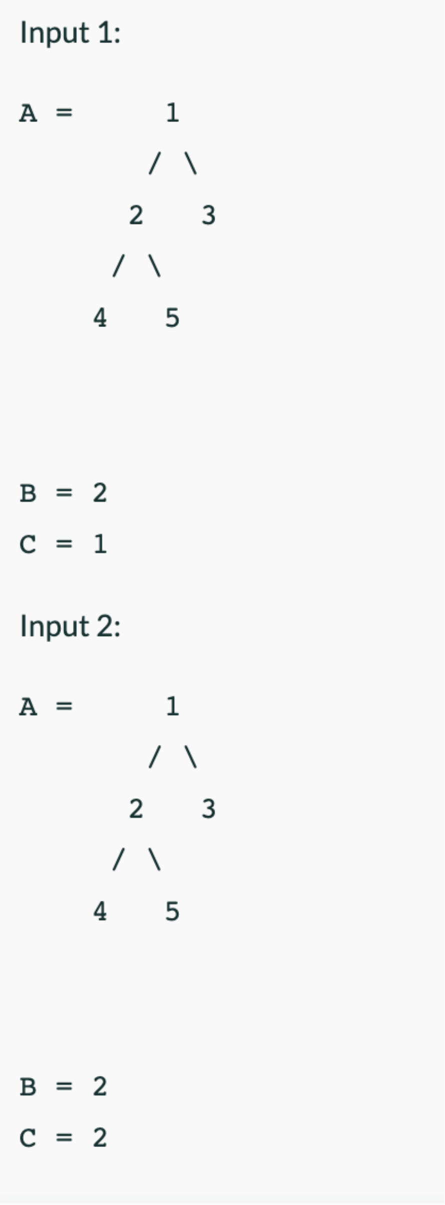
```
Input:
K = 3
arr[][] = {{1,2,3},{4,5,6},{7,8,9}}
Output: 1 2 3 4 5 6 7 8 9
Explanation:Above test case has 3 sorted
arrays of size 3, 3, 3
arr[][] = [[1, 2, 3],[4, 5, 6],
[7, 8, 9]]
The merged list will be
[1, 2, 3, 4, 5, 6, 7, 8, 9].
```

Nodes at a distance K from a target node



Given the root of a binary tree **A**, the value of a target node **B**, and an integer **C**, return an array of the values of all nodes that have a distance **C** from the target node.

You can return the answer in any order.



Input 1:

```
A =       1
         / \
        2   3
       / \
      4   5




B = 2
C = 1
```

Input 2:

```
A =       1
         / \
        2   3
       / \
      4   5




B = 2
C = 2
```

Output-1: [1, 4, 5]
Output-2: [3]