LAB1: Linear Array & Singly Linked List

[CO3]

Instructions for students:

- Complete the following methods on Array & Singly Linked List.
- You may use Java / Python to complete the tasks.
- DO NOT CREATE a separate folder for each task.
- If you are using JAVA, then follow the Java template
- If you are using PYTHON, then follow the python template.

NOTE:

- **YOU CANNOT USE ANY OTHER DATA STRUCTURE OTHER THAN ARRAYS or LINKED LIST.**
- **YOUR CODE SHOULD WORK FOR ANY VALID INPUTS. [Make changes to the Sample Inputs and check whether your program works correctly]**
- **LOOK OUT FOR 0th NODE Condition**

TOTAL TASKS: 5
TOTAL MARKS: 25

Linear Array

1. Merge Sorted Arrays

You are given two integer arrays arr1 and arr2, sorted in non-decreasing order (ascending order). Merge arr1 and arr2 into a single array sorted in non-decreasing order (ascending order) and return the new sorted array.

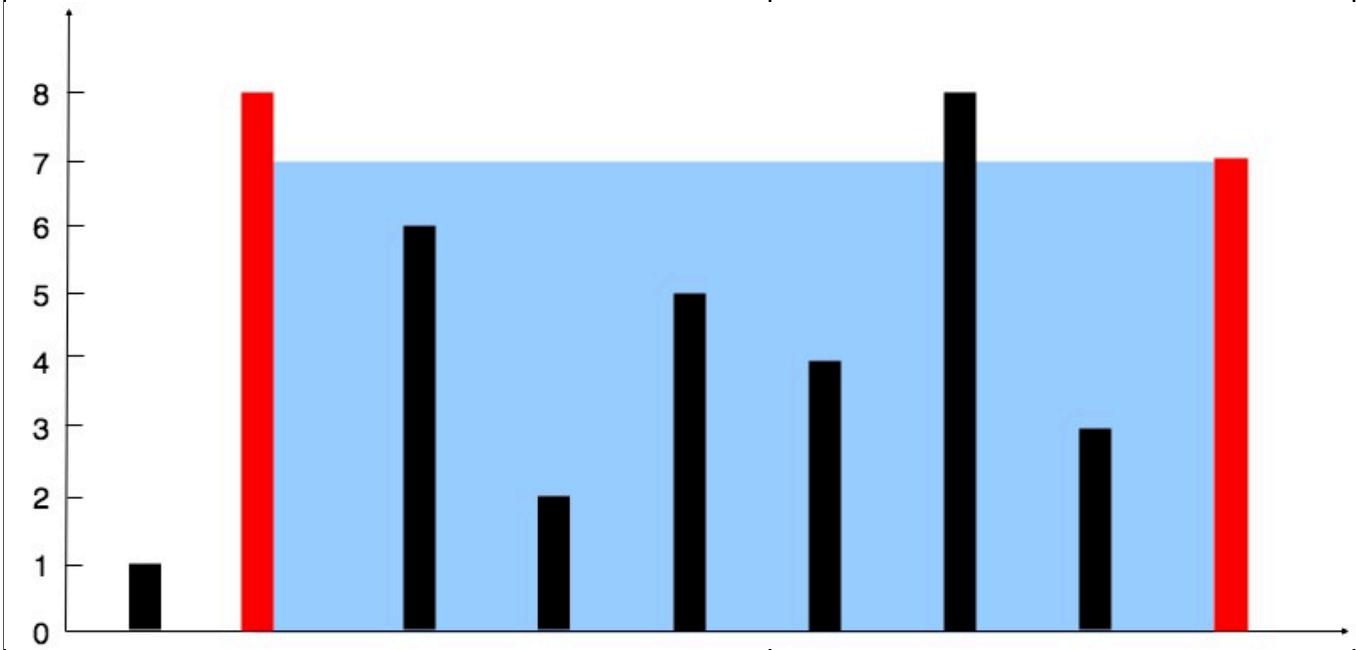
[Try to solve it in an optimized way]

Sample Given Arrays	Sample Returned Array
arr1 = [1,2,3] arr2 = [2,5,6]	[1,2,2,3,5,6]
arr1 = [1,3,5,11] arr2 = [2,7,8]	[1,2,3,5,7,8,11]

2. Container with Most Water

You are given an integer array called ***height*** of length n. You can imagine the values of the array as vertical lines drawn. If you imagine the y-axis and the x-axis then you can say the two endpoints of the **i**th vertical line are (i, 0) and (i, height[i]) .
Find such two lines that together with the x-axis form a container, such that the container contains the most water.
Return the maximum amount of water a container can store.

Note: that you may not slant the container.



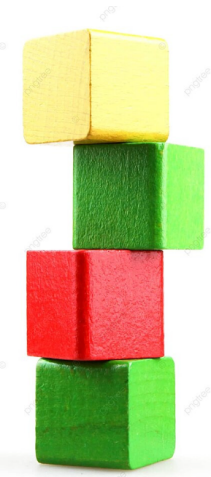
Sample Given Array	Sample Output
`height = [1,8,6,2,5,4,8,3,7]`	49

Explanation: Using the illustration you should be able to understand why we are choosing those two vertical lines. Now, if you look closely, the height of the rectangle is 7 and the length is 7 as well. Thus, the maximum area of water (blue colored area) is 49. Think how I calculated the height & length.

Singly Linked List

3. Building Blocks

Your twin and you are under an experiment where the amount of thinking similarities you two have is being observed. As per the experiment, you are given the same number of building blocks of different colors and are told to make a building using those blocks in two different rooms.



After the buildings are finished, the observers check whether the two buildings are the same based on the block colors. Now, you are the tech guy of that team and you are instructed to write a program that will output “Similar” or “Not Similar” given the two buildings. For fun, you decided to represent those buildings as a linked list!

NB: Red means a red block
Blue means a blue block
Yellow means a yellow block
Green means a green block.

Sample Input	Sample Output
building_1 = Red→ Green→ Yellow→ Red→ Blue→ Green building_2 = Red→ Green→ Yellow→ Red→ Blue→ Green	Similar
building_1 = Red→ Green→ Yellow→ Red→ Yellow→ Green building_2 = Red→ Green→ Yellow→ Red→ Blue→ Green	Not Similar
building_1 = Red→ Green→ Yellow→ Red→ Blue→ Green building_2 = Red→ Green→ Yellow→ Red→ Blue→ Green→ Blue	Not Similar

4. Assemble Conga Line

Have you ever heard the term conga line? Basically, it's a carnival dance where the dancers form a long line. Everyone holds the waist of the person in front of them and their waists are held in turn by the person to their rear, excepting only those in the front and the back. It kind of looks like this-

By now, you can quite understand the suitable data structure to represent a conga line. Now you are the choreographer of the Conga Dance in a Summer Festival. You wish to arrange the conga line **ascending** age wise and tell the participants to stand in a line likewise. Now as technical you are, can you write a method that will take the conga line and return True if everyone stands according to your instruction. Otherwise returns False.

Sample Input	Sample Returned Result
10 → 15 → 34 → 41 → 56 → 72	True
10 → 15 → 44 → 41 → 56 → 72	False

5. Sum of Nodes

You are given a Linked List, LL1, and an array, dist. Write a method **sum_dist(list, arr)** that takes a Linked List and an array as parameters. This method sums the node elements in the linked list that are away from the head by the elements in the array and returns the sum. Assume the Node class has only elem and next variable. **No need to write Node class and driver code.**

Sample Input	Sample Output	Explanation
LL1 = 10--> 16 --> -5 --> 9 --> 3 --> 4 dist = [2, 0, 5, 2, 8] Function Call: print(sum_dist(LL1, dist))	4	Node Element away from the head at distance 2 = -5 Node Element away from the head at distance 0 = 10 Node Element away from the head at distance 5 = 4 Node Element away from the head at distance 8 = Doesn't Exist, Considered as 0 The sum is: -5+10+4+-5+0 = 4