

Assignment-1

Ginjala Lohith Reddy(210050054)

May 9, 2023

1 What I have learnt

I have learnt many things in programming using python like different data structures which are **lists**, **dictionaries**, **tuples**, etc. I have learnt how to write a code in python with proper indentation. I have learnt about **slicing** of lists , how to access the elements in list, How to iterate a loop over a list. I have learnt how to write **loops**(while and for), nested loops and etc. I have learnt about **input()** function in python. I have learnt how to write functions using **def** in python. I have learnt how to write conditional statements using **if**, else if(**elif**) and **else**. I have also learnt about strings and different operations on strings. I have learnt different mathematical functions like **max** and **min** between two elements.

2 Leaders in Array

A function called **leader()** is defined in the provided code. It accepts a list **l** and size of list(**n**) as input and outputs a list of all the leaders in the input list. In order to hold the leaders, the function first initialises an empty list **lead**. The input list is then iterated over, with a new list being created for each entry that contains all the elements to the right of the current element using slicing method in lists. A leader is added to the list if the current element is greater than or equal to all the entries to its right. The list of leaders is then returned by the function.

3 Trapping Rain Water

A function named **water_trapped()** that receives as input an array of non-negative integers called **arr** and its length **n** and returns the maximum amount of water that may be trapped in the input array's bars. First, the function creates two n-sized arrays **l_max** and **r_max** both of which are initialised with "0". The largest element to the left and right of each element in the input array **arr** will be stored in these arrays. The **l_max** and **r_max** arrays are then filled by the function using two for loops. The greatest element to the left of each element in the input array is determined and stored in the corresponding index of the **l_max** array during the first loop, which iterates through the remaining elements of the array. In the second loop, the final element of the input array is used as the starting point, and the remaining

elements are iterated through in reverse order, with the maximum element to the right of each element being calculated and stored at the corresponding position of the **r_max** array. Finally, the function calculates the amount of water that can be trapped in between the bars by iterating over the input array and summing up the minimum of **l_max** and **r_max** at each index, subtracted by the value of the element at that index. The **result** is returned.

4 Minimum Platforms

A function called **min_platforms()** which accepts two arrays, **arr** and **dep** as inputs, representing the times of trains' **arrival** and **departure**, and returns the bare minimum number of platforms needed to fit all the trains at the station. The function sets the variable **pf**, which stands for the quantity of platforms currently required, to **1**, as its initial value. Additionally, it sets a variable called **min_pf** to **1** to serve as the minimum number of platforms required. The **arr** and **dep** arrays are then iterated through simultaneously by the function using a while loop. The number of platforms required drops by one if the current train's arrival time is earlier than the preceding train's departure time since one of the previously occupied platforms is now empty. A new platform is required for the current train and the required number of platforms rises by one if the arrival time of the current train is less than or equal to the departure time of the preceding train. The minimum number of platforms required is updated if the number of platforms required right now exceeds the number of platforms required initially. Finally **min_pf** value is returned.

5 Reverse a String

A function called **reverse_string()** that takes a string **x** as input and returns its reverse. The function iterates through each character of the string in **reverse** order using a for loop and appends each character to an empty string **l**. Once all the characters have been iterated through, the function returns the reversed string **l**.

6 Palindrome String

The `palindrome()` function, which takes a **string** `x` as input and returns **1** if the string is a palindrome (reads the same forward as it does backward) and **0** otherwise. As the function iterates through the characters of the input string in **reverse** order using a for loop, it appends each character to the newly created empty string `l`. The function returns **1**, indicating that the input string is a **palindrome**, if the reversed string `l` equals the input string `x`. If the supplied string is **not a palindrome**, it returns **0**, which means that it is not.