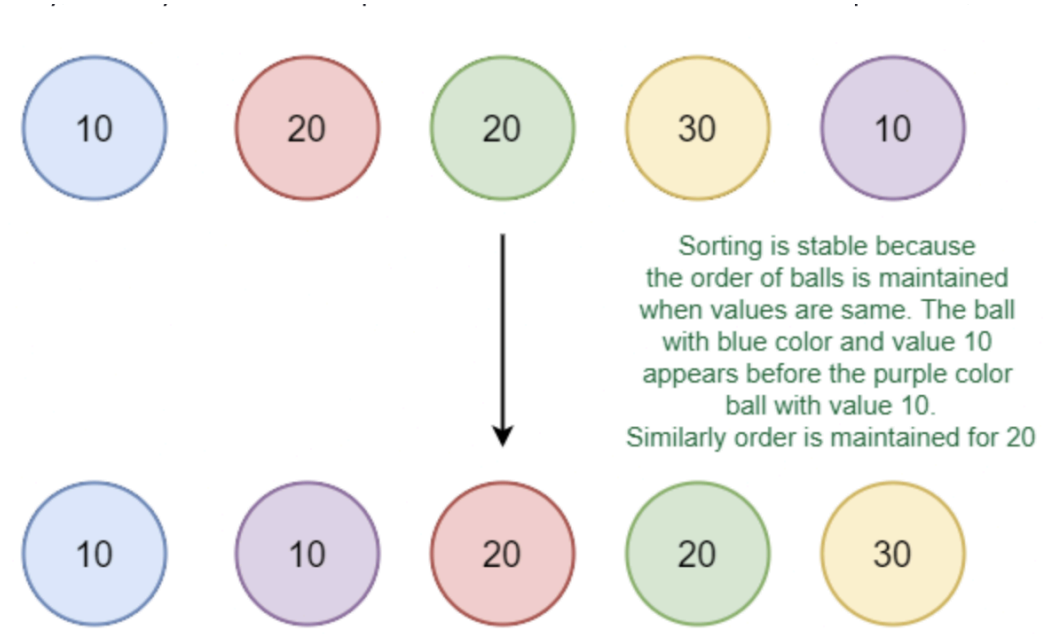
AGENDA

Sorting

- Self -
    - Bubble sort
    - Selection sort
    - Insertion sort
    - Quick sort
    - Cycle sort
    - Radix sort
    - Bucket sort
- Stability in sorting algorithms
- Sort a Big File
- Merge sort
- Merge 3 sorted arrays
- Union of 2 sorted arrays
- Minimum Platforms

Stability in Sorting Algorithms

A sorting algorithm is said to be stable if two objects with equal keys appear in the same order in sorted output as they appear in the input data set



Example of stable sort

Where stable sorting algorithms are useful?

Consider the following dataset of Student Names and their respective class sections.

$(Dave, A)(Alice, B)(Ken, A)(Eric, B)(Carol, A)$

If we sort this data according to name only, then it is highly unlikely that the resulting dataset will be grouped according to sections as well.

$(Alice, B)(Carol, A)(Dave, A)(Eric, B)(Ken, A)$

So we might have to sort again to obtain the list of students section-wise too. But in doing so, if the sorting algorithm is not stable, we might get a result like this:

$(Carol, A)(Dave, A)(Ken, A)(Eric, B)(Alice, B)$

The dataset is now sorted according to sections, but not according to names. In the name-sorted dataset, the tuple (Alice, B) was before (Eric, B) , but since the sorting algorithm is not stable, the relative order is lost. If on the other hand, we used a stable sorting algorithm, the result would be:

$(Carol, A)(Dave, A)(Ken, A)(ALice, B)(Eric, B)$

Here the relative order between different tuples is maintained. It may be the case that the relative order is maintained in an Unstable Sort but that is highly unlikely.

Sort a Big File

Suppose you have a 20 GB file with one string present per line. How would you sort the file?

Approach-1 : Use any built-in sorting algorithm to sort the strings lexicographically - simply O(n * log(n)) where n is the number of lines (or strings) present in the file.

But for this, you will have to load 20 GB of data into memory!

- Is this feasible?

Think more!!

Merge Sort

Given an array arr[], its starting position l and its ending position r. Sort the array using the merge sort algorithm.

Examples:

Input: arr[] = [4, 1, 3, 9, 7]
Output: [1, 3, 4, 7, 9]

Input: arr[] = [10, 9, 8, 7, 6, 5, 4, 3, 2, 1]
Output: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

Input: arr[] = [1, 3 , 2]
Output: [1, 2, 3]

Constraints:
1 <= arr.size() <= 10^5
1 <= arr[i] <= 10^5

Merge 3 Sorted Arrays

Given three sorted arrays **A**, **B** and **C** of size **N**, **M** and **P** respectively. The task is to merge them into a single array which must be sorted in increasing order.

Example 1:

Input:
N = 4, A[] = [1 2 3 4]
M = 5, B[] = [1 2 3 4 5]
P = 6, C[] = [1 2 3 4 5 6]
Output: 1 1 1 2 2 2 3 3 3 4 4 4 5 5 6
Explanation: Merging these three sorted arrays, we have:
1 1 1 2 2 2 3 3 3 4 4 4 5 5 6.

Example 2:

Input:
N = 2, A[] = [1 2]
M = 3, B[] = [2 3 4]
P = 4, C[] = [4 5 6 7]
Output: 1 2 2 3 4 4 5 6 7

Minimum Platforms

You are given the arrival times **arr[]** and departure times **dep[]** of all trains that arrive at a railway station on the same day. Your task is to determine the minimum number of platforms required at the station to ensure that no train is kept waiting.

At any given time, the same platform cannot be used for both the arrival of one train and the departure of another. Therefore, when two trains arrive at the same time, or when one arrives before another departs, additional platforms are required to accommodate both trains.

Assume that if one train arrives and another departs at the same time, then arrival operation occurs before the departure.

Examples:

Input: arr[] = [900, 940, 950, 1100, 1500, 1800], dep[] = [910, 1200, 1120, 1130, 1900, 2000]
Output: 3
Explanation: There are three trains during the time 9:40 to 12:00. So we need a minimum of 3 platforms.

Input: arr[] = [900, 1235, 1100], dep[] = [1000, 1240, 1200]
Output: 1
Explanation: All train times are mutually exclusive. So we need only one platform

Input: arr[] = [1000, 935, 1100], dep[] = [1200, 1240, 1130]
Output: 3
Explanation: All 3 trains have to be there from 11:00 to 11:30