8 days:

**Day 5**

**Sorting and Searching Algorithms**

Bubble sort : O(n2)

Best case : already sorted 🡪only one pass needed.

1,2,3,4,5

Average case : random input 🡪 multiple pass required depending upon the value.

5,3,1,2,4

Worst case : reversed sorted 🡪 maximum swaps required. If element already in sorted in ascending order to make descending order is worst case.

5,4,3,2,1

1. Selection sort : Selector sort is another simplest sorting technique. Which use comparison- based sorting algorithm.

It works by dividing input list into two parts.

A sorted sub-list and unsorted sub-list. Initially the sorted sub-list is empty. While then unsorted sub-list contains all elements of the input list or data. We find minimum or maximum element iterate through unsorted list or data. Swap with first unsorted elements. Then expand sorted sub-list.

Best case : n(n-1)/2 : if array already sorted. Selection sort still scan or check entire array to find the minimum for each position. 0(n2)

1,2,3,4,5

Average case : - 0(n2)

4,1,3,2,5

Worst case : - 0(n2)

5,4,3,2,1

**Insertion sort:** This sort build an array one item or data at time by taking each element or data and placing it in the correct position relative to be already sorted part in array.

Step

1. Assume the first element is sorted.
2. Take the next element or data and insert in into the sorted part with condition.
3. Shift element or data as need to make the space.
4. This process repeats till the data get sort ie Ascending or descending order.

**Merge Sort:** Merge sort is an efficient, comparison-based and divide and conquer paradigm algorithms. It works by divided our input into two parts (halves) and sorting each half recursively and then merges the sorted halves to produce a sorted array.

29,10,14,37,14

Divided into two parts

Sort [29,10] : 1st part

Sort [14,37,14] part