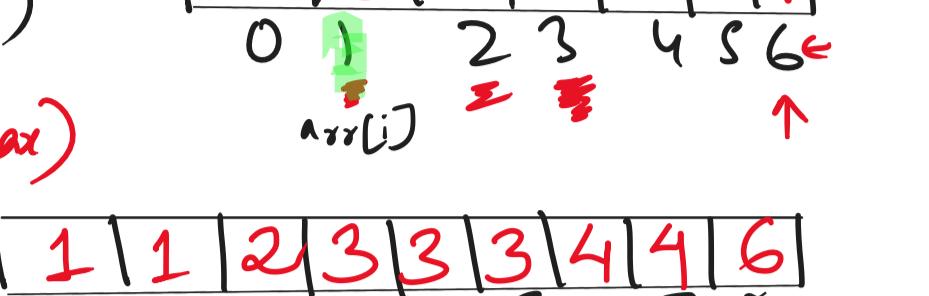
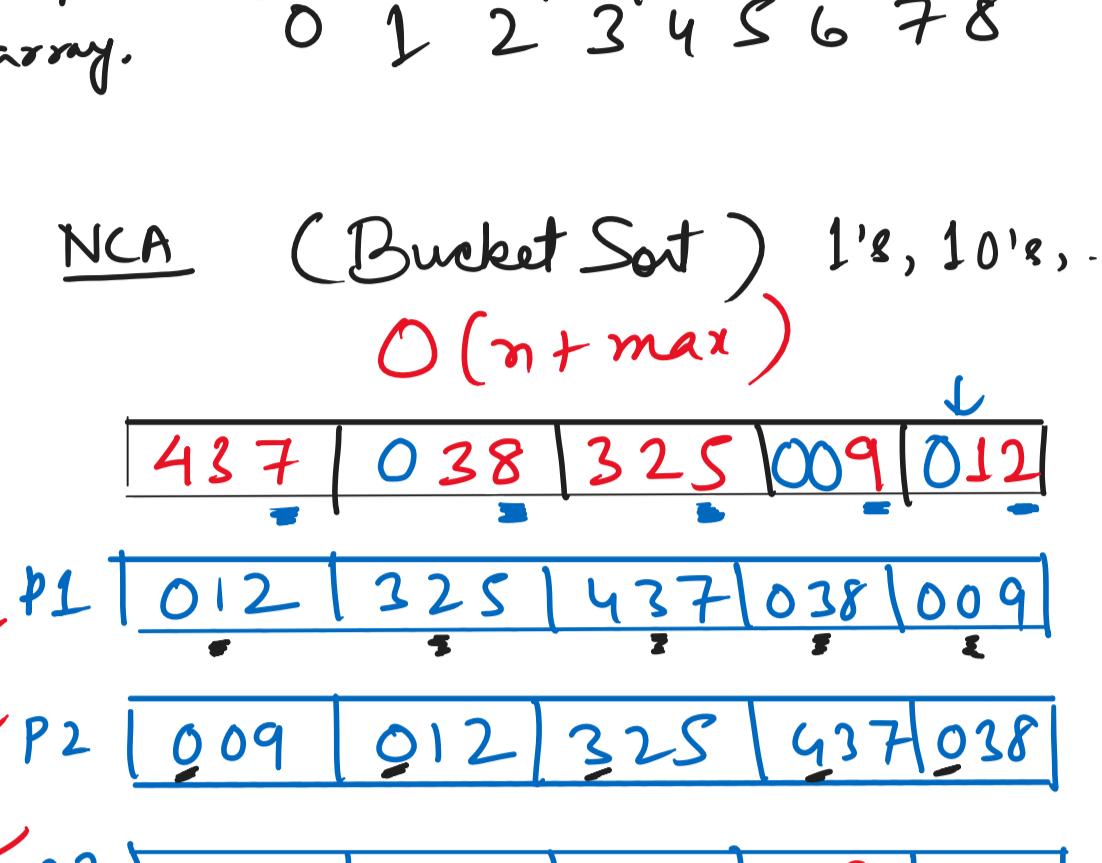


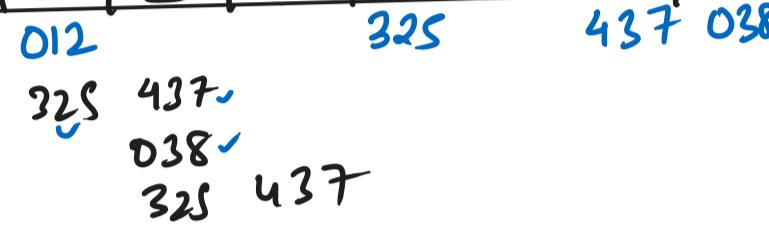
* Non Comparison Sorting Algorithms :-

Count Sort Algorithm :-

- * Single digit whole numbers (0-9)
- Step 1: Find the max = 6
- Step 2: Create a count array 0 to max [0-6]
- Step 3: Calculate the frequency of each element.
- Step 4: Calculate cumulative count of the array.
- Step 5: Create the o/p array
- Step 6: Start from the end of the array.

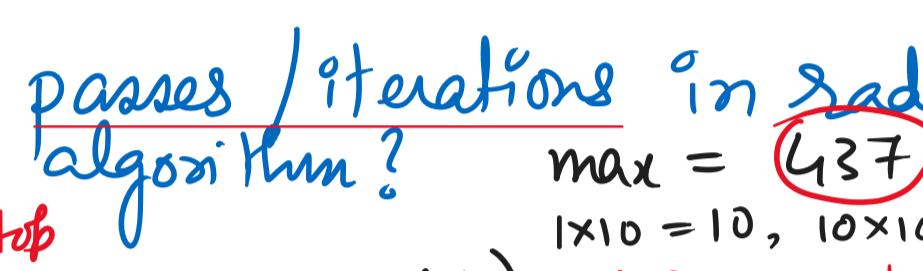
Input →  (len=9) ↓ ↓ L → R



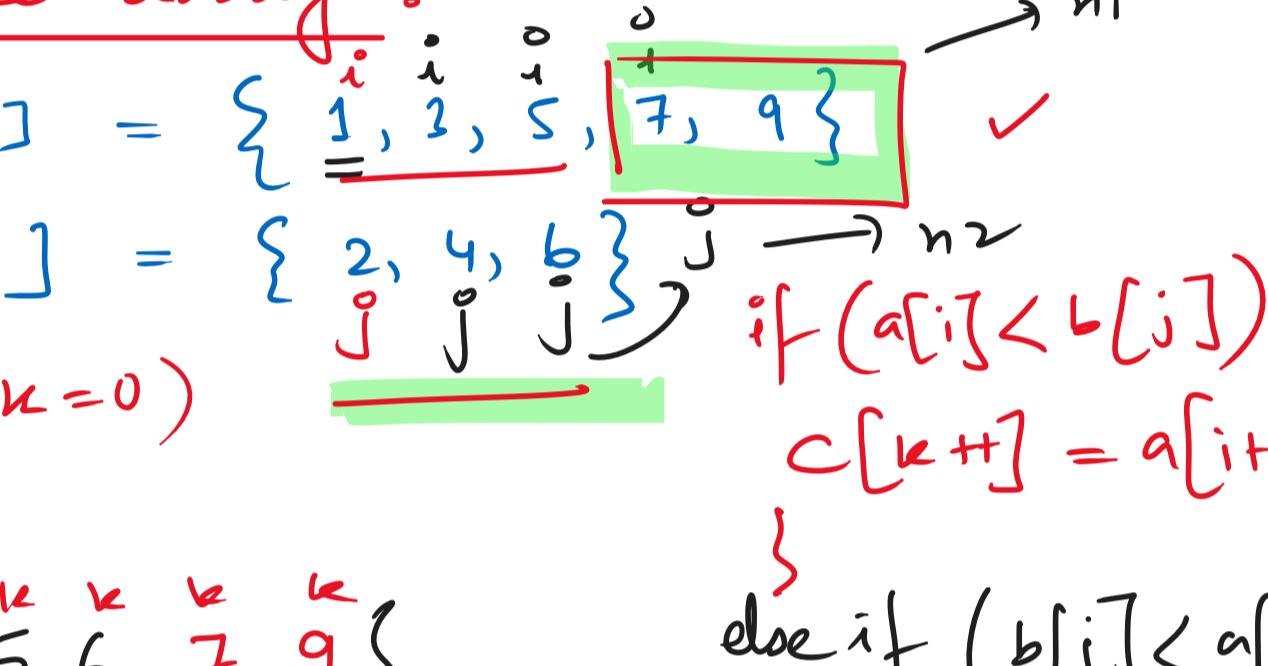
Output → 

* Radix Sort Algorithm :- NCA (Bucket Sort) 1's, 10's, ... O(n + max)

- 1. Single digits
- 2. Multi digits
- 3. Constant length strings "john" "root"


 P_1 | 012 | 325 | 437 | 038 | 009
 P_2 | 009 | 012 | 325 | 437 | 038
 P_3 | 009 | 012 | 038 | 325 | 437

- Step 1: Find the max = 437
3 digits = 3 passes
- Step 2: Create (0-9)
10 buckets

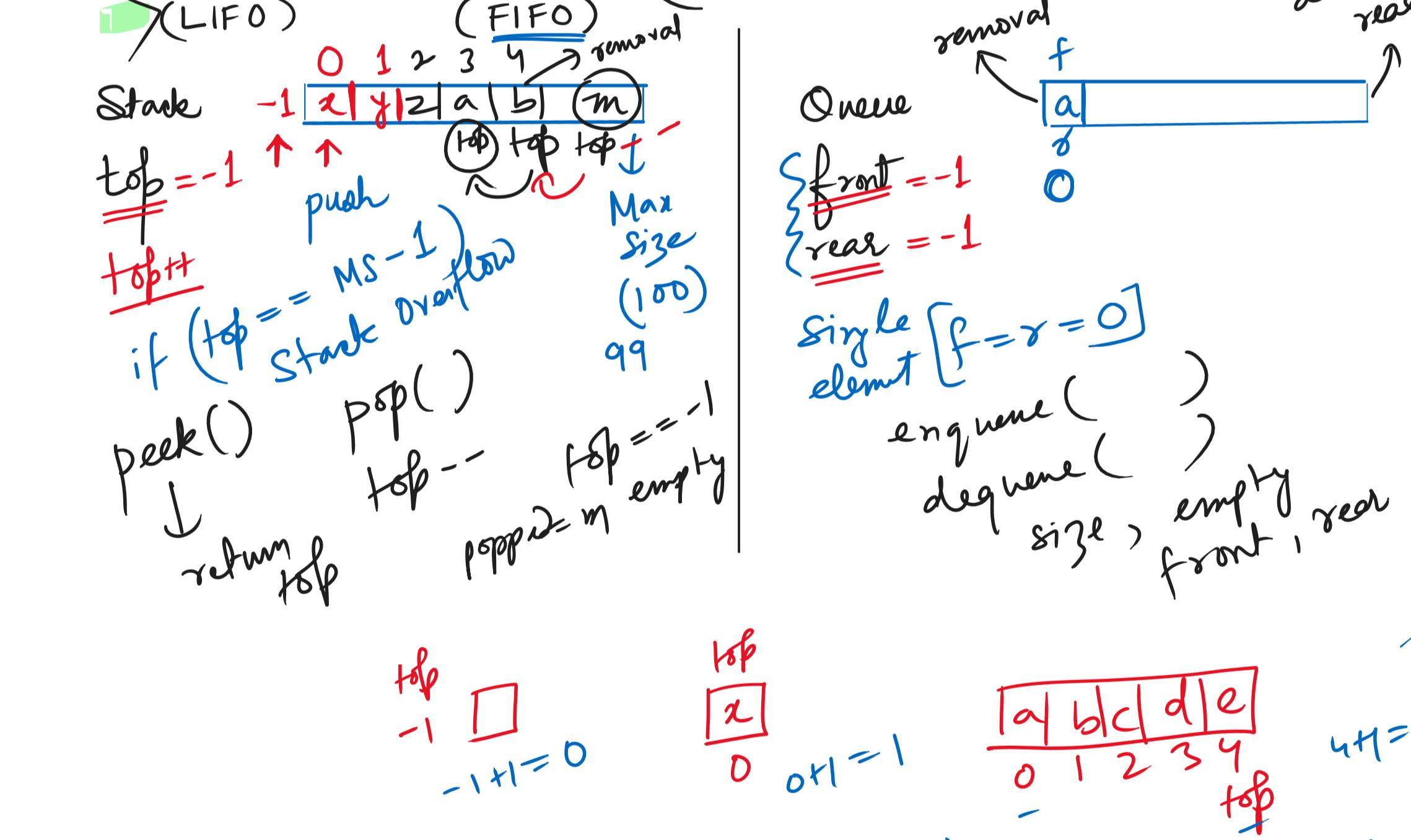
Step 3: 1's then 10's and so on.


* Radix & Count Sort are brothers. Each pass (iteration) of radix sort calls the count sort method to handle the single digit element.

*** Q. How are the number of passes / iterations in radix sort controlled in the algorithm?

for (int exp=1; max/exp > 0; exp *= 10) {
 countSort(arr, exp, size);
 max = 437
 $1 \times 10 = 10, 10 \times 10 = 100$
 $100 \times 10 = 1000$
 $437/1 = 437$
 $437/10 = 43$
 $437/100 = 4$
 $437/1000 = 0$

Merge two sorted arrays :-



* Merge Sort Algorithm: (Divide & Conquer Algorithm) (Recursion)

($\log n$) division Space Complexity $O(n)$ Single Element Sorted $\frac{n}{2} \times \log n$ $n \log n$

Time Complexity $O(\log n) + O(n) = \frac{6}{2} = 3$ ($n \log n$) = 3

Space Complexity $O(1)$ (merges) $O(n)$

Time Complexity $O(n \log n)$ n element merge

Space Complexity $O(1)$ (merges) $O(n)$

Time Complexity $O(n \log n)$ n element merge

Space Complexity $O(1)$ (merges) $O(n)$

Time Complexity $O(n \log n)$ n element merge

Space Complexity $O(1)$ (merges) $O(n)$

Time Complexity $O(n \log n)$ n element merge

Space Complexity $O(1)$ (merges) $O(n)$

Time Complexity $O(n \log n)$ n element merge

Space Complexity $O(1)$ (merges) $O(n)$

Time Complexity $O(n \log n)$ n element merge

Space Complexity $O(1)$ (merges) $O(n)$

Time Complexity $O(n \log n)$ n element merge

Space Complexity $O(1)$ (merges) $O(n)$

Time Complexity $O(n \log n)$ n element merge

Space Complexity $O(1)$ (merges) $O(n)$

Time Complexity $O(n \log n)$ n element merge

Space Complexity $O(1)$ (merges) $O(n)$

Time Complexity $O(n \log n)$ n element merge

Space Complexity $O(1)$ (merges) $O(n)$

Time Complexity $O(n \log n)$ n element merge

Space Complexity $O(1)$ (merges) $O(n)$

Time Complexity $O(n \log n)$ n element merge

Space Complexity $O(1)$ (merges) $O(n)$

Time Complexity $O(n \log n)$ n element merge

Space Complexity $O(1)$ (merges) $O(n)$

Time Complexity $O(n \log n)$ n element merge

Space Complexity $O(1)$ (merges) $O(n)$

Time Complexity $O(n \log n)$ n element merge

Space Complexity $O(1)$ (merges) $O(n)$

Time Complexity $O(n \log n)$ n element merge

Space Complexity $O(1)$ (merges) $O(n)$

Time Complexity $O(n \log n)$ n element merge

Space Complexity $O(1)$ (merges) $O(n)$

Time Complexity $O(n \log n)$ n element merge

Space Complexity $O(1)$ (merges) $O(n)$

Time Complexity $O(n \log n)$ n element merge

Space Complexity $O(1)$ (merges) $O(n)$

Time Complexity $O(n \log n)$ n element merge

Space Complexity $O(1)$ (merges) $O(n)$

Time Complexity $O(n \log n)$ n element merge

Space Complexity $O(1)$ (merges) $O(n)$

Time Complexity $O(n \log n)$ n element merge

Space Complexity $O(1)$ (merges) $O(n)$

Time Complexity $O(n \log n)$ n element merge

Space Complexity $O(1)$ (merges) $O(n)$

Time Complexity $O(n \log n)$ n element merge

Space Complexity $O(1)$ (merges) $O(n)$

Time Complexity $O(n \log n)$ n element merge

Space Complexity $O(1)$ (merges) $O(n)$

Time Complexity $O(n \log n)$ n element merge

Space Complexity $O(1)$ (merges) $O(n)$

Time Complexity $O(n \log n)$ n element merge

Space Complexity $O(1)$ (merges) $O(n)$

Time Complexity $O(n \log n)$ n element merge

Space Complexity $O(1)$ (merges) $O(n)$

Time Complexity $O(n \log n)$ n element merge

Space Complexity $O(1)$ (merges) $O(n)$

Time Complexity $O(n \log n)$ n element merge

Space Complexity $O(1)$ (merges) $O(n)$

Time Complexity $O(n \log n)$ n element merge

Space Complexity $O(1)$ (merges) $O(n)$

Time Complexity $O(n \log n)$ n element merge

Space Complexity $O(1)$ (merges) $O(n)$

Time Complexity $O(n \log n)$ n element merge

Space Complexity $O(1)$ (merges) $O(n)$

Time Complexity $O(n \log n)$ n element merge

Space Complexity $O(1)$ (merges) $O(n)$

Time Complexity $O(n \log n)$ n element merge

Space Complexity $O(1)$ (merges) $O(n)$

Time Complexity $O(n \log n)$ n element merge

Space Complexity $O(1)$ (merges) $O(n)$

Time Complexity $O(n \log n)$ n element merge

Space Complexity $O(1)$ (merges) $O(n)$

Time Complexity $O(n \log n)$ n element merge

Space Complexity $O(1)$ (merges) $O(n)$

Time Complexity $O(n \log n)$ n element merge

Space Complexity $O(1)$ (merges) $O(n)$

Time Complexity $O(n \log n)$ n element merge

Space Complexity $O(1)$ (merges) $O(n)$

Time Complexity $O(n \log n)$ n element merge

Space Complexity $O(1)$ (merges) $O(n)$

Time Complexity $O(n \log n)$ n element merge

Space Complexity $O(1)$ (merges) $O(n)$

Time Complexity $O(n \log n)$ n element merge

Space Complexity $O(1)$ (merges) $O(n)$

Time Complexity $O(n \log n)$ n element merge

Space Complexity $O(1)$ (merges) $O(n)$

Time Complexity $O(n \log n)$ n element merge

Space Complexity $O(1)$ (merges) $O(n)$

Time Complexity $O(n \log n)$ n element merge

Space Complexity $O(1)$ (merges) $O(n)$

Time Complexity $O(n \log n)$ n element merge

Space Complexity $O(1)$ (merges) $O(n)$

Time Complexity $O(n \log n)$ n element merge

Space Complexity $O(1)$ (merges) $O(n)$

Time Complexity $O(n \log n)$ n element merge

Space Complexity $O(1)$ (merges) $O(n)$

Time Complexity $O(n \log n)$ n element merge

Space