

## Code

```
int partition ( arr, s, e ) {
```

```
    pivot = arr[e];     $O(1)$ 
```

```
    i = s-1;    1
```

```
    for ( j = s; j < e ; j++ ) {     $O(N)$ 
```

```
        if ( arr[j] < pivot ) {     $O(1)$ 
```

```
             $O(1)$ 
```

```
            i++;
```

```
             $O(1)$ 
```

```
            swap ( i, j ) // swap elements  
                           on index i  
                           & j;
```

```
        }
```

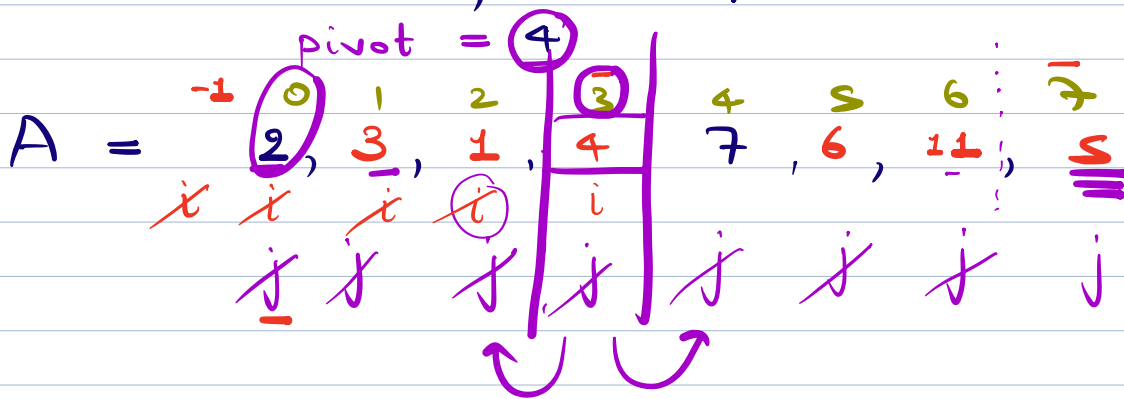
```
    }
```

```
     $O(1)$  swap ( i+1, e );
```

```
     $O(1)$  return ( i+1 );
```

```
}
```

$s = 0, e = 7$



first

random  $\Rightarrow$  s, e

$$T.C. = \underline{\underline{O(N)}}$$

Q

Given an array of size N.

A[i]  $\Rightarrow$  0, 1, 2  $\Rightarrow$  Info.

Sort the array.

i/p = 1, 0, 0, 2, 1, 0, 2

o/p  $\Rightarrow$  0, 0, 0, 1, 1, 2, 2

Sol<sup>n</sup>

1) Sort the array  $\Rightarrow$  ds/ms  
 $\Rightarrow$   $N \log N$

2) Count the no. of 0's, 1's & 2's

space

$\downarrow$  0  $\Rightarrow$  3

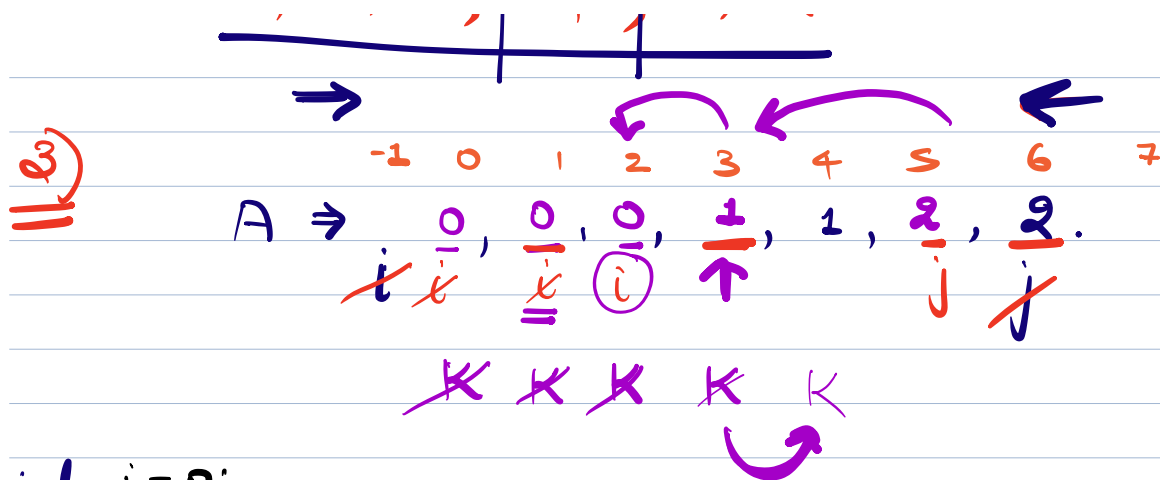
1  $\Rightarrow$  2

2  $\Rightarrow$  2

$\leftarrow$  0, 0, 0, | 1, 1 |  $\rightarrow$  2, 2

T.C. =  $O(N)$

S.C. =  $O(3)$



```

int i = 0;
for (i = 0; i < N; i++) {
    if (A[i] != 0) {
        break;
    }
}

```

// 0's left

```

int j = 0;
for (j = n-1; j > 0; j--) {
    if (A[j] != 2) {
        break;
    }
}

```

// 2's right

2

```

K = i;
while (K <= j) {
    if (A[K] == 0) {

```

```

        swap(i, K); // Ele on
        i++;        i, & K
        K++;        index.
    }
}

```

else if ( $A[k] == \underline{2}$ ) {

swap(k, j);

j--;

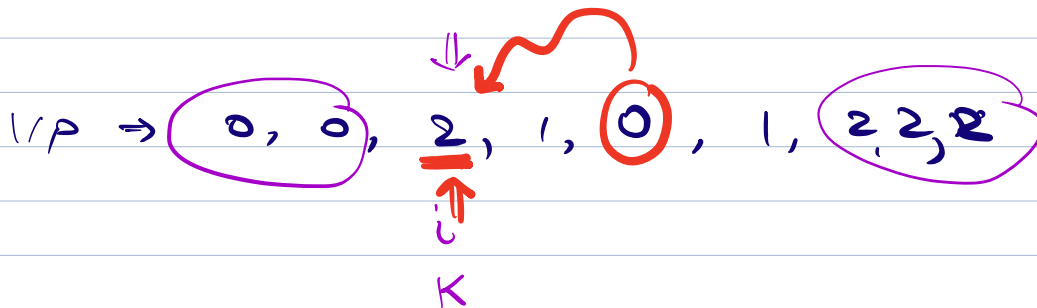
~~k++;~~ X  
if ( $A[k] == 0$ ) {

swap(i, k);

i++;

k++;

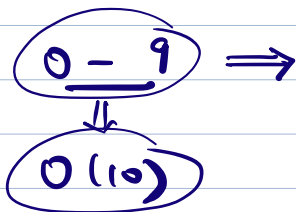
k



Limit  $\Rightarrow$  3 distinct elements.

---

Count Sort  $\uparrow$



$\Rightarrow$  Traverse & maintain frequency.

Range of ele  $\Rightarrow [0, 10]$

$\Downarrow$   
Array?? Map??

$A \Rightarrow [50, \underline{10}, 50, \underline{1000}, 50, \underline{20}]$

0 0 0      1    0    1    0    2      1  
0, 1, 2, ... 10, ... 20, ... 50, ... 1000

$\Downarrow$ 

10	1
20	1
50	2
1000	1

 $\rangle$  HashMap  $\times$   
Ordered??  
 $\times$

Tree Map  $\Rightarrow$  Ins.  
 $\log N$       Retrieval  
 $\log N$

Iterate  $\Rightarrow$   $O(N)$

Creation  $\Rightarrow$   $N \log N$

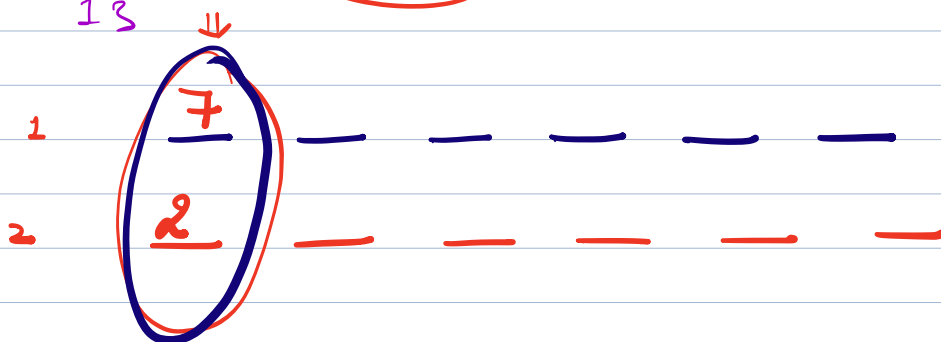
✓	5	1	2
✓	3	6	9
✓	3	6	2
✓	4	3	1
✓	5	1	6
✓	6	4	5
✓	9	3	8

$0-9 \times$

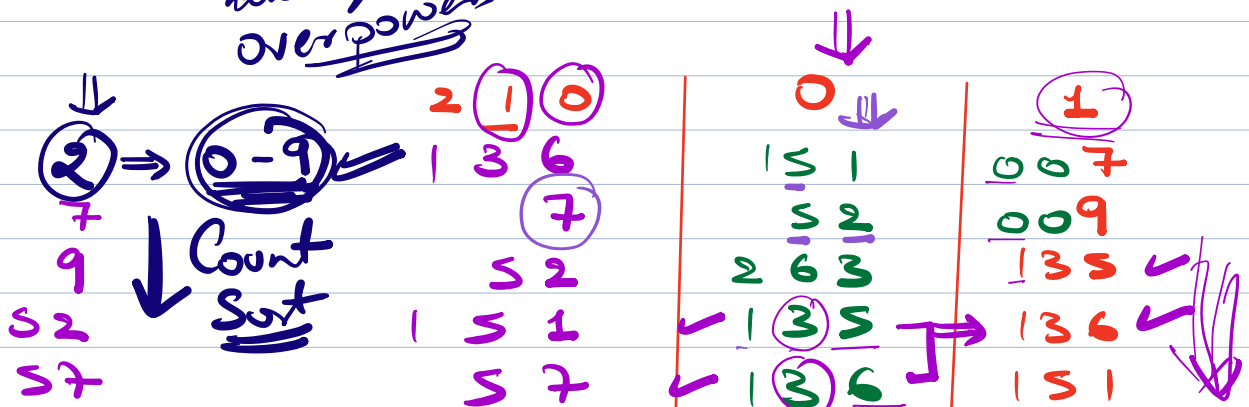
## Raddix Sort

$\times \rightarrow \downarrow \downarrow \leftarrow \checkmark$   
 $\begin{array}{c} 3 \ 1 \\ 1 \ 3 \end{array}$

$31$   
 $13$   
 $10 > 31$   
 $13 > 31$  ✓



MSB  
 always  
 overpowers



135  
136  
151  
263

9  
135  
263

07  
57  
09

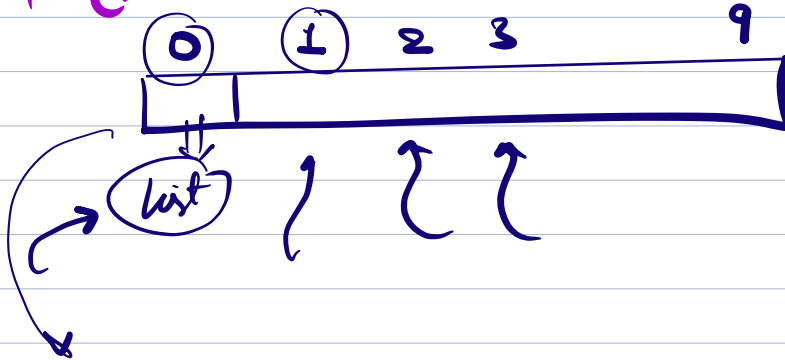
052  
057  
263



19  
28  
37  
46  
59  
63  
72  
81  
90  
18  
19

INT MAX  
214 748 3647

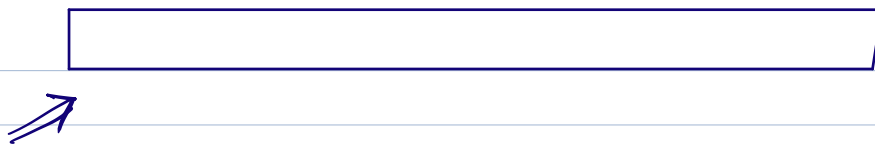
T.C. =



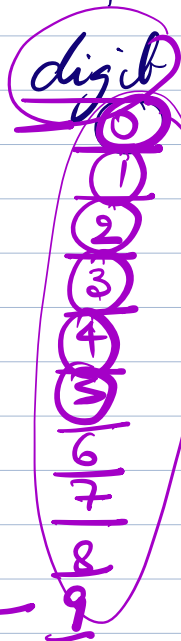
! 361, 432, 12, 78, 500, 112,  
62, 121, 240, 59, 139, 86.

0

9



Hash Map



List of Elements.

500, 240  
361, 121  
432, 12, 112, 62

$O(N)$

78  
59, 139

$$\begin{aligned}
 & \underline{\underline{86}} \\
 & \begin{array}{r} 36 \\ \downarrow \\ (361/10) \cdot 10 \end{array} \quad \begin{array}{c} \text{361} \\ \text{78} \end{array} \quad 361 \cdot 10 \\
 & \begin{array}{c} \text{361} \\ \text{100} \end{array} \cdot 10 = 3
 \end{aligned}$$

$$\begin{aligned}
 & \underline{\underline{N}} \quad \underline{\underline{N-1}} \quad \dots \quad \underline{\underline{3}} \quad \underline{\underline{2}} \quad \underline{\underline{1}} \\
 & \begin{array}{c} \text{N} \\ \downarrow \\ \left( \frac{N}{10^{N-1}} \right) \cdot 10 \end{array} \quad \begin{array}{c} \left( \frac{N}{100} \right) \cdot 10 \quad \left( \frac{N}{10} \right) \cdot 10 \quad N \cdot 10 \\ \left( \frac{N}{10^2} \right) \cdot 10 \end{array}
 \end{aligned}$$

H.W. Count Sort / Radix Sort



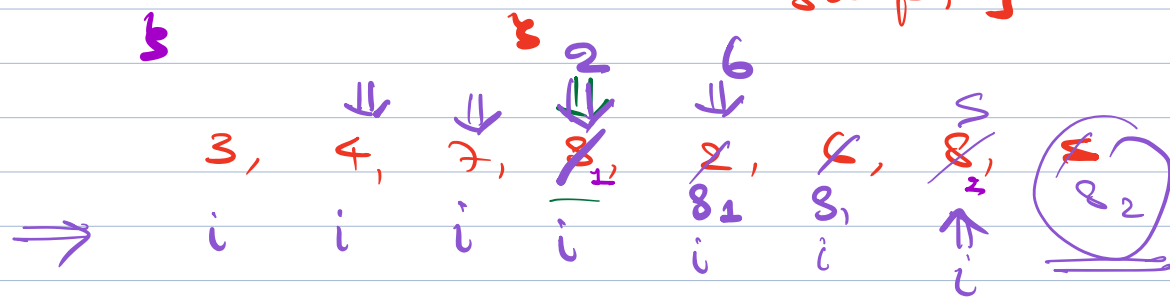
# Stable Sort

## 1) Bubble Sort

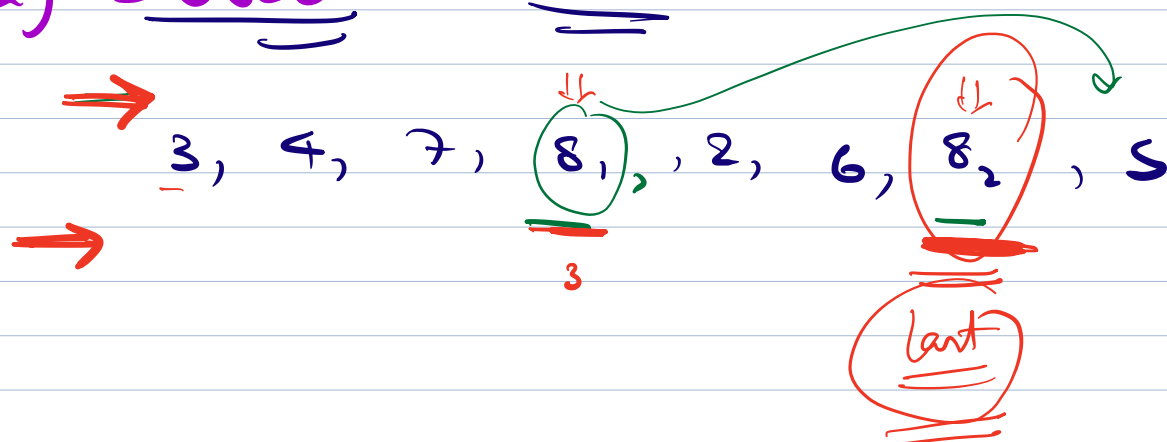
for ( $k=0$ ;  $k \leq n-1$ ;  $k++$ ) {

for ( $i=0$ ;  $i < n-k$ ;  $i++$ ) {

if ( $A[i+1] < A[i]$ ) {  
swap; }



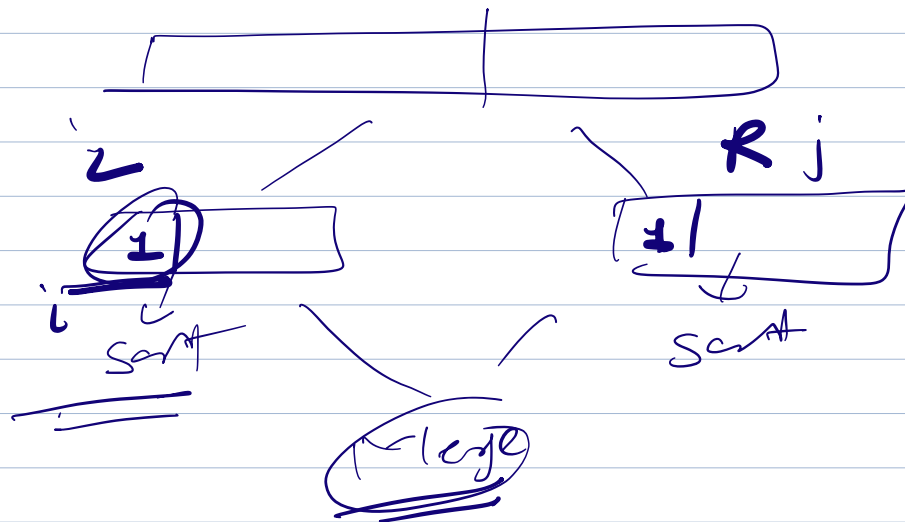
## 2) Selection Sort



### 3) Insertion Sort



### 4) Merge Sort



$$\text{if } (A_1[i] \leq A_2[j]) \{$$

H.W.  
5

Quick Sort  $\Rightarrow$

④ 3, 1, 6, 4, 4, 4, 8, 9, 11

⑥ Raddix Sort