Sorting is an arrangement of data in particular order on the basis of some parameter.

Ex:- 1 $\{2, 3, 9, 12, 17, 19\}$

Parameter → Inc magnitude of elements.

Ex:- 2 $\{19, 6, 5, 2, -1, -19\}$

Parameter → Dec:- magnitude of elements

Quiz 1

$\{1, 13, 9, 6, 12\}$

Count of Factors → $1, 2, 3, 4, 6$

Sorted.

Q:-1 Given an array of N integers, minimise the cost to empty given array where cost of removing an element is equal to sum of element left in arr

Ex:-

A = {2, 1, 4}	Cost
{2, 1, 4}	7
{2, 4}	6
{2}	2
	15

A = {2, 1, 4}	Cost
{2, 1, 4}	7
{2, 1}	3
{1}	1
	11

Quiz 2

	Cost
{4, 6, 1}	11
{4, 13}	5
{1}	1
	17

Quiz 3

	Cost
{3, 5, 1, -3}	6
{3, 1, -3}	1
{1, -3}	-2
{-3}	-3
	2

Approach

Generalisation

0	1	2	3
a	b	c	d

Remove a	a + b + c + d
Remove b	b + c + d
Remo c	c + d
Remov d	d
	a + 2b + 3c + 4d

$a \geq b \geq c \geq d$

Pseudo Code

```
int calculate_cost(int arr[]) {
    reverse_sort(arr);  // N lg N

    int ans = 0;
    for (int i=0; i<N; i++) {
        ans += (i+1) * arr[i];
    }
    return ans;
}
```

N

A - {3, 5, 1, -3}

{5, 3, 1, -3}

0 1 2 3

i

ans = ~~0~~
5
+6
+3
-12

2

T.C → O(N Log N)

S.C → O(N)

Q-2. Given an array of distinct elements of size N, find the count of noble integers.

Note:- arr[i] is noble if count of elements smaller than arr[i] = arr[i]

Ex:- {1, -5, 3, 5, -10, 4}

0 1 2 3 4 5

ans = 3.

Quiz 4

{-3, 0, 2, 5}

0 1 2 5

ans = 1.

Brute Force

```
int findNoble(int arr[]) {
    int ans = 0;
    for (i = 0; i < N; i++) {
        int cont = 0;
        for (j = 0; j < N; j++) {
            if (arr[j] < arr[i]) {
                cont++;
            }
        }
        if (cont == arr[i]) {
            ans++;
        }
    }
    return ans;
}
```

Update cont

TC → $O(N^2)$

S.C → $O(1)$

Ex:

0	1	2	3	4	5
-3	-1	0	1	2	5

Optimisation

```
int findNoble(int arr) {
    sort(arr);                    // N lg N
    int ans = 0;
    for (i = 0; i < N; i++) {
        if (i == arr[i])          // N
            ans++;
    }
    return ans;
}
```

m + N

N lg N

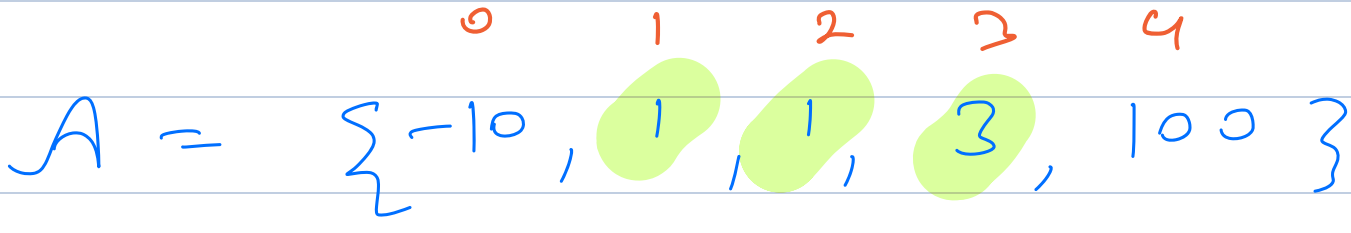
N

T.C. → O(N lg N)

S.C → O(N)

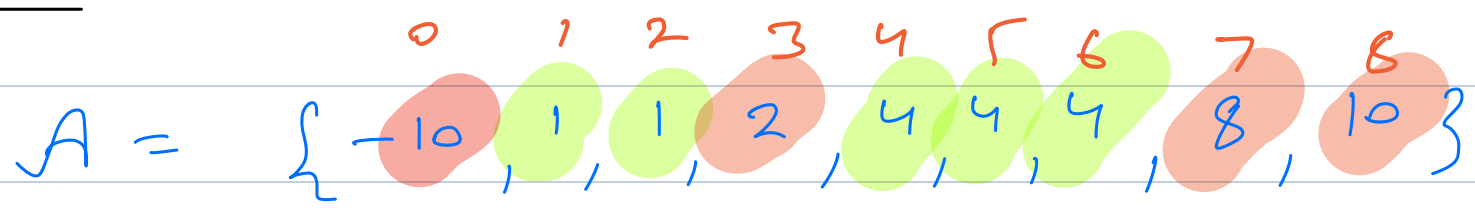
Q:- What of the elements are not distinct ?,

Quiz 5



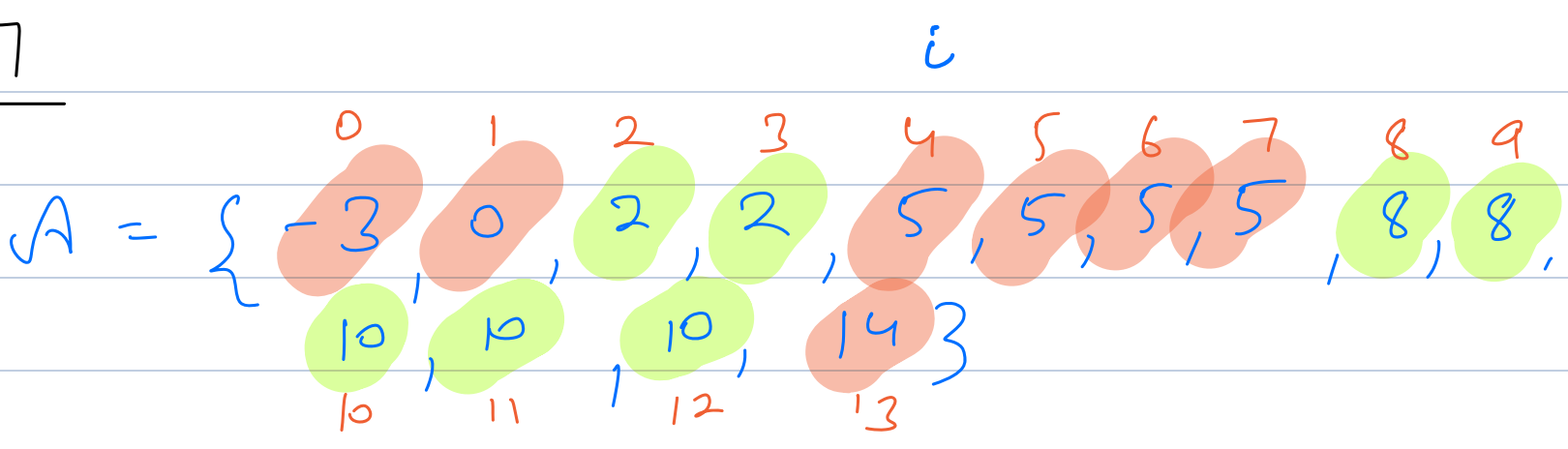
ans = 3.

Quiz 6



ans = 5.

Quiz 7



Solution

1. // If the curent elent is same as previous elent, the total no. of smaller elents of curent elent = those of previous elent.

2. // If the curent elent is not same as previous elent, c will tell me the no. of smaller elents than A[i];

Code → TODO - H.W.

8:25

Selection Sort

0 1 2 3

Arr : {5, 6, 4, 2, 3}.

i

Code

```
for (i=0; i<N; i++) {
    minIndex = i;
    for (j=i+1; j<N; j++) {
        if (arr[j] < arr[minIndex]) {
            minIndex = j;
        }
    }
    swap(arr[minIndex], arr[i]);
}
```

T.C → $O(N^2)$

S.C → $O(1)$

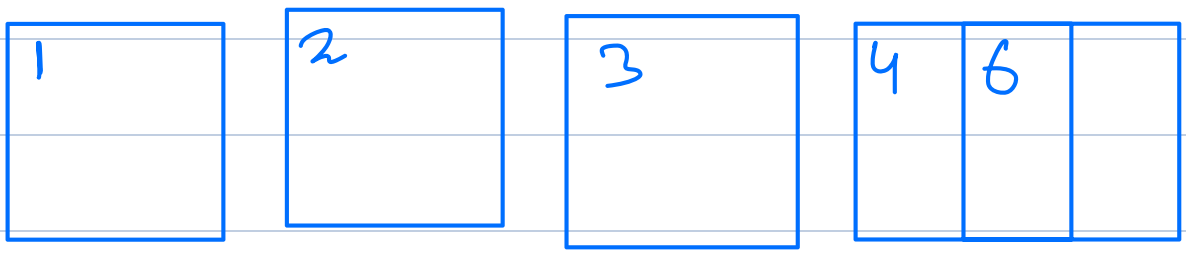
i = 0; minIndex = 0; j = 1, 2, 3

0 1 2 3

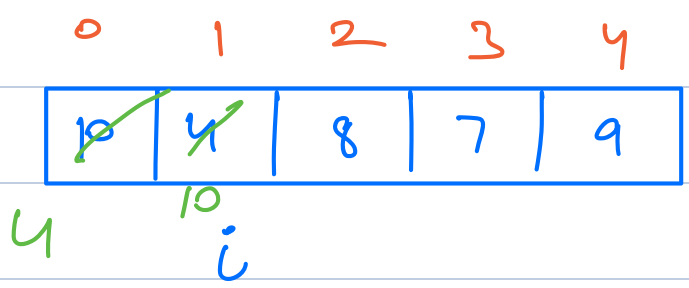
{5, 6, 4, 2, 3}

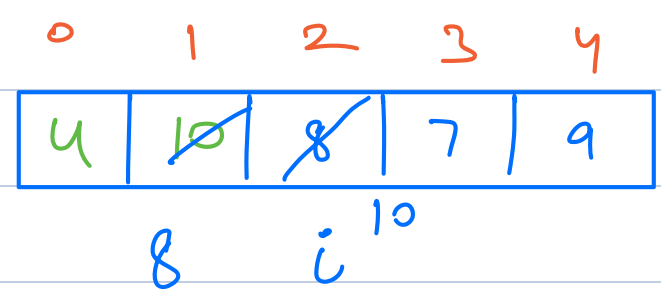
i = 1; minIndex = 1; j = 2, 3

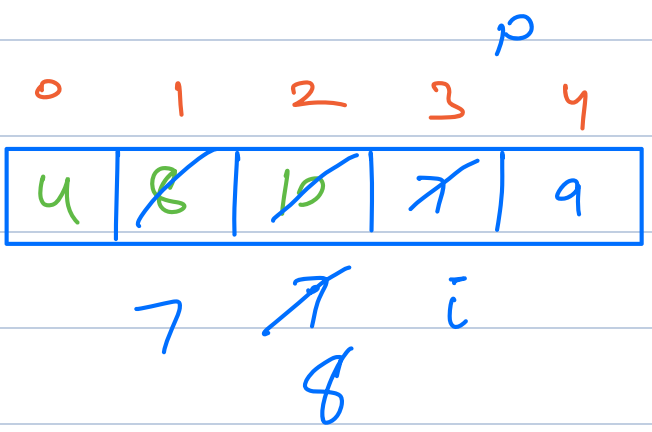
Insertion Sort

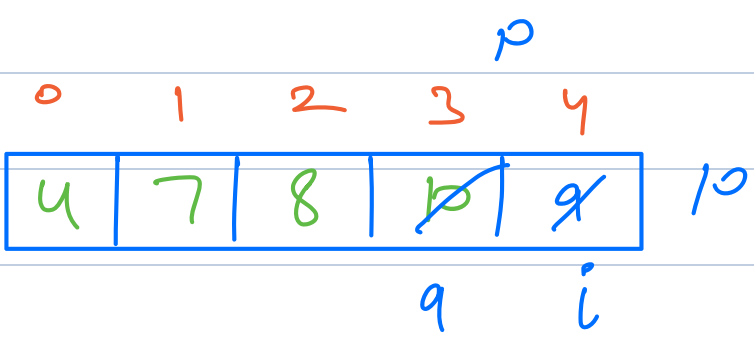


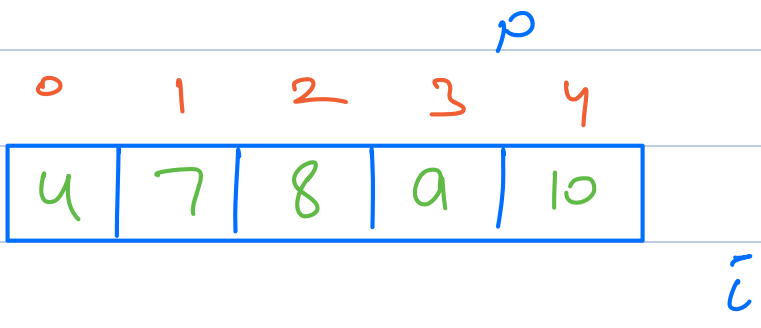
Ex!-





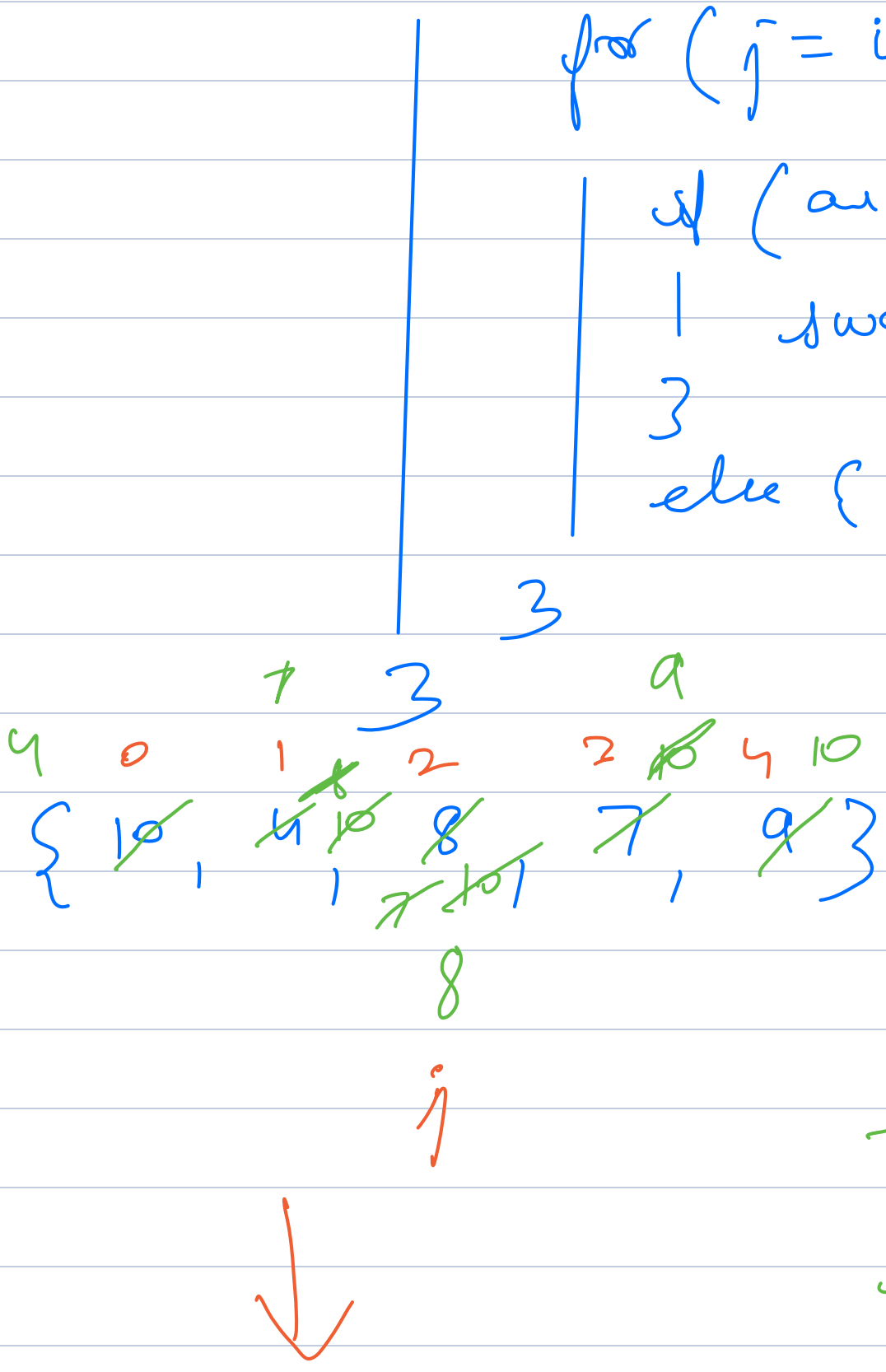






Code:

```
for (i=1; i<N; i++){
    for (j=i-1; j>=0; j--){
        if (arr[j] > arr[j+1]){
            swap(arr[j], arr[j+1]);
        }
        else { break; }
    }
}
```



T.C → $O(N^2)$.

S.C → $O(1)$

{4, 7, 8, 9, 10}.

In place Sorting → Where SC is $O(1)$

Insertion Sort

↓

{4, 7, 8, 9, 10}.
 ↑
 i

O(N) iterations for already sorted arr.

↓

Best Case T.C.

TODO 7. What is best case T.C. for selection sort ?

8, 1, 1