Sorting

TABLE OF CONTENTS

1. Understand sorting
2. Few problems on sorting
3. 2 sorting algorithms
    - 3.1 Selection Sort
    - 3.2 Insertion Sort

Sorting

Arrangement of data in a particular order basis a certain parameter

{2, 3, 9, 17, 21} ascending order on the basis of number
↓
small to big

{19, 6, 5, -1, -100} descending on basis of number
↓
big to small

{1, 7, 4} basis number of factors
1 2 3

{1, 13, 9, 6, 12}
1 2 3 4 6

Why sorting?

Organization of data

$\underbrace{\text{C++, Java, Python}}_{\text{inbuilt sort function}}$

arr = {2, 3, 1, 5, 4}

arr.sort() sort(arr)

{1, 2, 3, 4, 5}

Question (Elements Removal)

Given N elements, at every step remove an array element.

Cost to remove an element = Sum of array of elements present in an array

Find minimum cost to remove all elements.

NOTE : First add the cost of removal and then remove it.

remove the biggest guy first

arr - [2,1,4]	7
[1,4]	5
[4]	4
x	16

	7
[2,1]	3
[1]	1
x	11

{4,6,1}	11
4,1	5
1	1
x	17

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

lets say { a,b,c,d } in desc order

0 1 2 3

a+b+c+d

b+c+d

c+d

d

$1 \times a + 2b + 3c + 4d$

1) sort

2) reverse

```
ans = 0
for( i : 0 -> n-1) {
        ans += (i+1) x a[i]
}
return ans
```

$n\log n + n + n$

$n\log n$

TC: $O(n\log n)$

3, 5, 1, -3

5, 3, 1, -3

0 1 2 3

$1 \times 5 + 2 \times 3 + 3 \times 1 + 4 \times -3$

$5 + 6 + 3 - 12 = \underline{2}$

Question (Noble Integers) { Distinct data }

Given N array elements, calculate number of noble integers.

An element ele in arr [ ] is said to be noble if { count of smaller elements = ele itself }

arr - [1 , -5 , 3 , 5 , -10 , 4]

less 2 1 3 5 0 4

arr - [-3 , 0 , 2 , 5]

0 1 2 3

	2	4	6	8	10
idx	0	1	2	3	4
less	0	1	2	3	4

Brute: For each elem, count how many smaller in a nested loop.

TC: $O(n^2)$

Better sol^n ⇒ sort the array

After sorting less = idx

Ques asks a[i] = less

a[i] = i

1) sort

```
count = 0
for( i: 0 → n-1) {
        if( a[i] == i)
                count ++
}
```

sort ⇒ nlogn time

Question (Noble Integers) : { Data can repeat }

arr - [-10 , 1 , 1 , 3 , 100] ans = 3

idx	0	1	2	3	4
less	0	1	1	3	4

arr - [-10 , 1 , 1 , 2 , 4 , 4 , 4 , 8, 10] ans = 5

idx	0	1	2	3	4	5	6	7	8
less	0	1	1	3	4	4	4	7	8

arr - [-3 , 0 , 2 , 2 , 5 , 5 , 5 , 5 , 8 , 8 , 10 , 10 , 10 , 14]

idx	0	1	2	3	4	5	6	7	8	9	10	11	12	13
less	0	1	2	2	4	4	4	4	8	8	10	10	10	13

</> Code

1) If same number ⇒ less value is same
2) If diff number ⇒ less value = idx

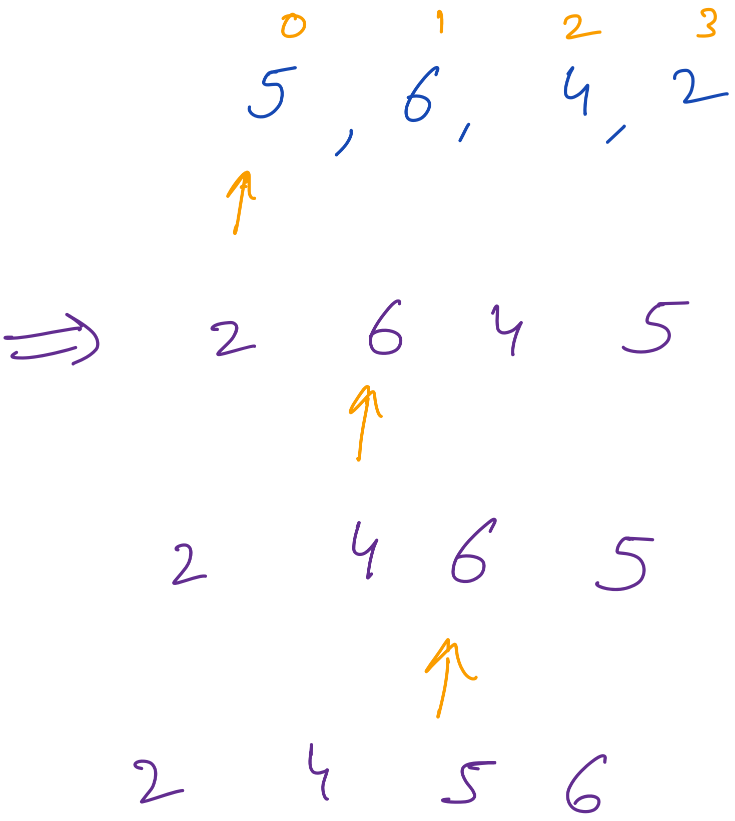
```
less = 0        count = 0        if(a[0] == 0)
                                    count++

for( i: 1 → n-1) {
    if( a[i] != a[i-1])    }  condition-2
        less = i
    if ( a[i] == less)     }  check for noble integer
        count++
}
```

Selection Sort

idea : Select the minimum element and send that elements to correct position by swapping.

Select smallest element one by one

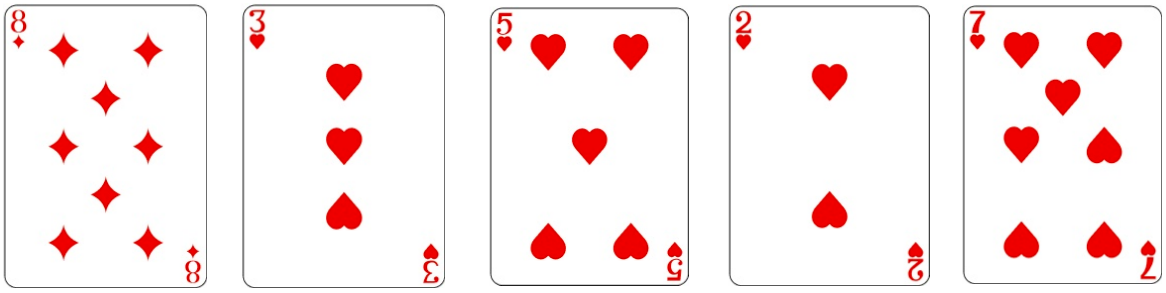


</> Code

```
for(i: 0 → n-1) {
    smallest_val_idx = i
    for(j: i+1 → n-1) {
        if ( arr[j] < arr[smallest_val_idx])
            smallest_val_idx = j
    }
    swap(arr[i], arr[smallest_val_idx]
}
```

TC: $O(n^2)$

Insertion Sort (Arrangement of playing cards)



2 4 5 7 8 10 add 3

2 3 4 5 7 8 10 add 6

2 3 4 5 6 7 8 10

{ 4, 2, 1, 7, 10, 5 }

⇑

sorted ⇒ 1 2 4 5 7 10

TC: $O(N^2)$

</> Code

TODO

1) New array

2) Iterate on unsorted array

    2.1 For elem, iterate on sorted array to find correct pos.

    2.2 Insert at correct pos.