Average PSP ⇒ > 75 %.

PSP < 60

20 %.
40 %.
60 %.

→ Just attending the class is not enough.

60 ≤ PSP < 80

< 80 %. ✓

Att > 70-80 %.

< 60 %.

60 - 80 %.

Att > 80-90 %. ✓

PSP ≥ 80 %.
Healthy

Assignment ↙

⇓

Solution ✓
Pseudo Code ✓

⇓

Easy ⇒ __ mins
Medium ⇒ __ mins
Hard ⇒ __ mins

} Improve this day by day.

Additional

Sorting

Arrangement of data in a specific order based on some parameter

↙ Ascending ↘ Descending.

A = { 2, 3, 9, 12, 17, 19 }

↓

Sorted in Asc. based on value of elements

A = { 19, 6, 5, 2, -1, -19 }

↓

Sorted in Desc. based on value of elements

A = { 1, 13, 9, 6, 12 }

Count of factors = { 1, 2, 3, 4, 6 } → Sorted on the basis of no. of factors.

Why is sorting important ??

Given an integer array of size N. Remove all elements one by one.

Cost of removing an element = Sum of values of all elements present in the array before removing the element.

find min cost of removing all elements.

$A = \{ \underset{0}{2}, \underset{1}{1}, \underset{2}{4} \}$

Approach 1

$2 \Rightarrow 7 \Rightarrow \{1, 4\}$
$1 \Rightarrow 5 \Rightarrow \{4\}$
$4 \Rightarrow 4 \Rightarrow \{\}$

16

Approach 2

$4 \Rightarrow 7 \Rightarrow \{2, 1\}$
$2 \Rightarrow 3 \Rightarrow \{1\}$
$1 \Rightarrow 1 \Rightarrow \{\}$

11

Min cost to remove all ele. from

$\{4, 6, 1\}$

$6 \Rightarrow 11 \Rightarrow \{4, 1\}$
$4 \Rightarrow 5 \Rightarrow \{1\}$
$1 \Rightarrow 1 \Rightarrow \{\}$

17

Min cost to remove all ele. from

$\{3, 5, 1, -3\}$

$5 \Rightarrow 6 \Rightarrow \{3, 1, -3\}$
$3 \Rightarrow 1 \Rightarrow \{1, -3\}$
$1 \Rightarrow -2 \Rightarrow \{-3\}$
$-2 \Rightarrow -3 \Rightarrow \{\}$

2

Approach

A = [a, b, c, d] →

	Cost	Array
Remove a	a + b + c + d	[b, c, d]
Remove b	b + c + d	[c, d]
Remove c	c + d	[d]
Remove d	d	[ ]

Total cost : $a + 2 \times b + 3 \times c + 4 \times d$.

↑ Max ↑ 2nd Max ↑ 3rd Max ……

⇓

Remove the elements in descending order.

1) Sort the element in descending order. ??

```
ans = 0;
for (i=0; i<N; i++) {          // O(N)
    ans = ans + (i+1) x A[i];
}
```

return ans;

T.C. = $O(N \log N + N)$;

$= O(N \log N)$

S.C. = _____

Given an integer array of size N. Find the count of Noble integers in the array. (All elements are distinct)

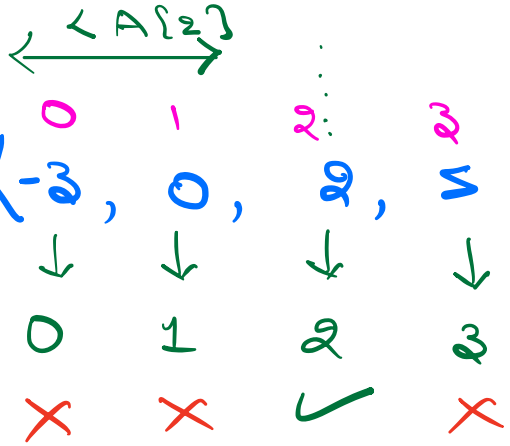
Noble integer → arr[i] is a noble integer if the count of elements less than arr[i] is equal to arr[i] itself.

	0	1	2	3	4	5
A =	1	-5	3	5	-10	4
Count of no. smaller than A[i].	2	1	3	5	0	4
	✗	✗	✓	✓	✗	✓

Ans = 3

NOTE: -ve elements can never be noble integers

as count can never be negative.



Ans = 1

Soln ⇒ Brute force

⇒ ∀ element ⇒ Iterate & find the count of smaller elements.

Code

```
ans = 0;
for (i=0; i<N; i++) {
    count = 0;
    for (j=0; j<N; j++) {
        if (A[j] < A[i]) {
            count++;
        }
    }
    if (count == A[i]) { ans++; }
}
```

find the count of elements smaller than A[i].

return ans;

T.C. = $O(N^2)$
S.C. = $O(1)$

2) Optimise Goal: Count of elements on left of any element on index i.

	0	1	2	3	4	5
A =	1	-5	3	5	-10	4

	0	1	2	3	4	5
SA =	-10	-5	1	3	4	5
	↓	↓	↓	↓	↓	↓
	0	1	2	3	4	5

for any element on index i

⇒ Count of element from [0, i-1]

= i

$\forall$ i ⇓ if (A[i] == i) { ⇒ Noble Integer }

T.C. = $O(N \log N)$

Given an integer array of size N. Find the count of Noble Integers.

$$A = \{ \underset{0}{0}, \underset{1}{1}, \underset{2}{1}, \underset{3}{2}, \underset{4}{4}, \underset{5}{4} \}$$

Quiz

$$A = \{ \underset{0}{-10}, \underset{1}{1}, \underset{2}{1}, \underset{3}{3}, \underset{4}{100} \}$$

0 1 1 3 4

Quiz

$$A = \{ \underset{0}{-10}, \underset{1}{1}, \underset{2}{1}, \underset{3}{2}, \underset{4}{4}, \underset{5}{4}, \underset{6}{4}, \underset{7}{8}, \underset{8}{10} \}$$

Count → 0 1 1 2 4 4 4 7 8

Code

```
int findNobelIntegers(A[], N) {
    int ans = 0;
    sort(A);
    int count = 0;
    if (A[0] == 0) { ans++; }

    for (i = 1; i < N; i++) {
        if (A[i] != A[i-1]) {
            count = i;
        }
```

```
        if (count == A[i]){
            ans++;
        }
    }
    return ans;
}
```

T.C. = $O(N \log N)$

S.C. =

Sorting

⇒ Inbuilt sorting functions.

Java ⇒ `Collections.sort(ArrayList);`
`Arrays.sort(Array);`

C++ ⇒ `sort(a.begin(), a.end());`

Python ⇒ `A.sort()`
`A = sorted(A)`

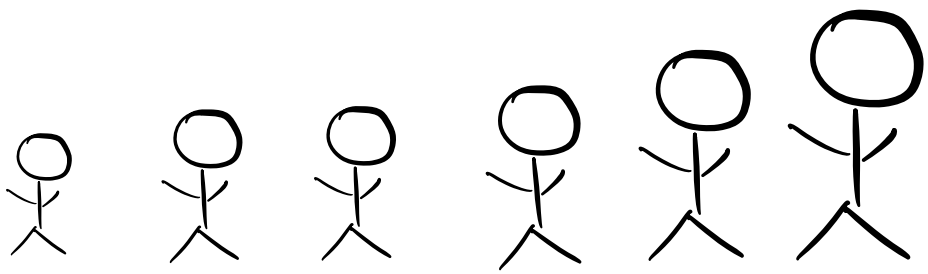
JS ⇒ A.sort()

T.C. = $O(N \log N)$
S.C. = $O(\log N)$

H.W. ⇒ figure out function call to sort the array in decreasing order.

Selection Sort

Scenario



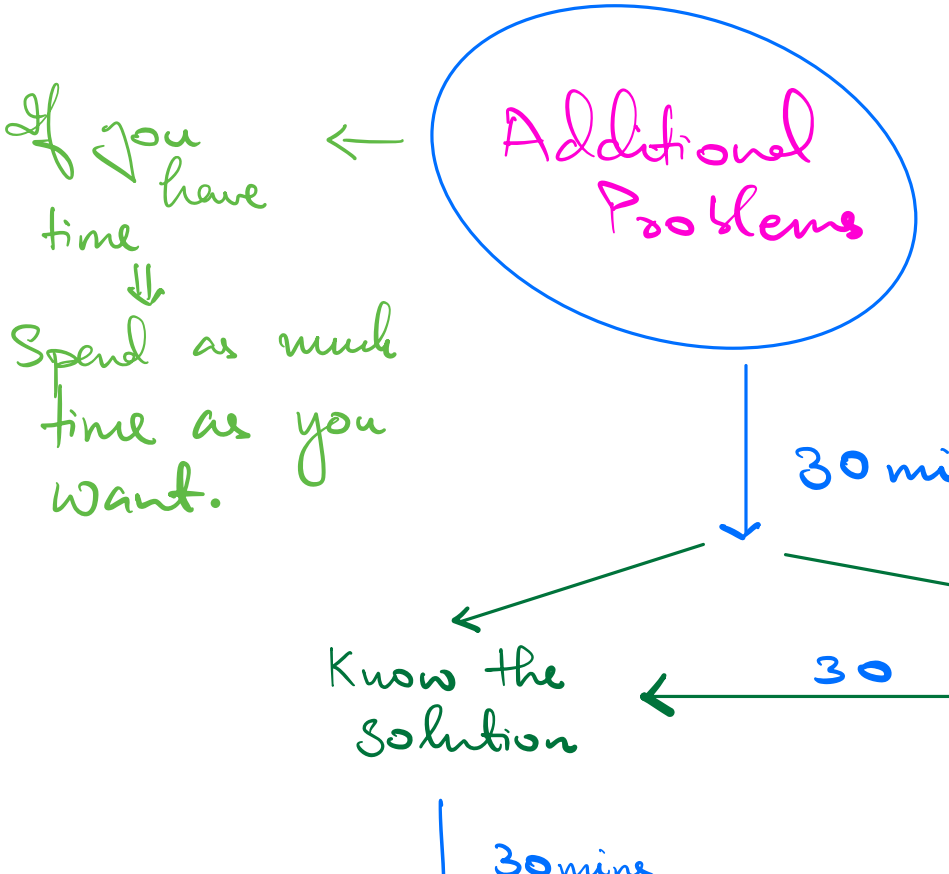
Code

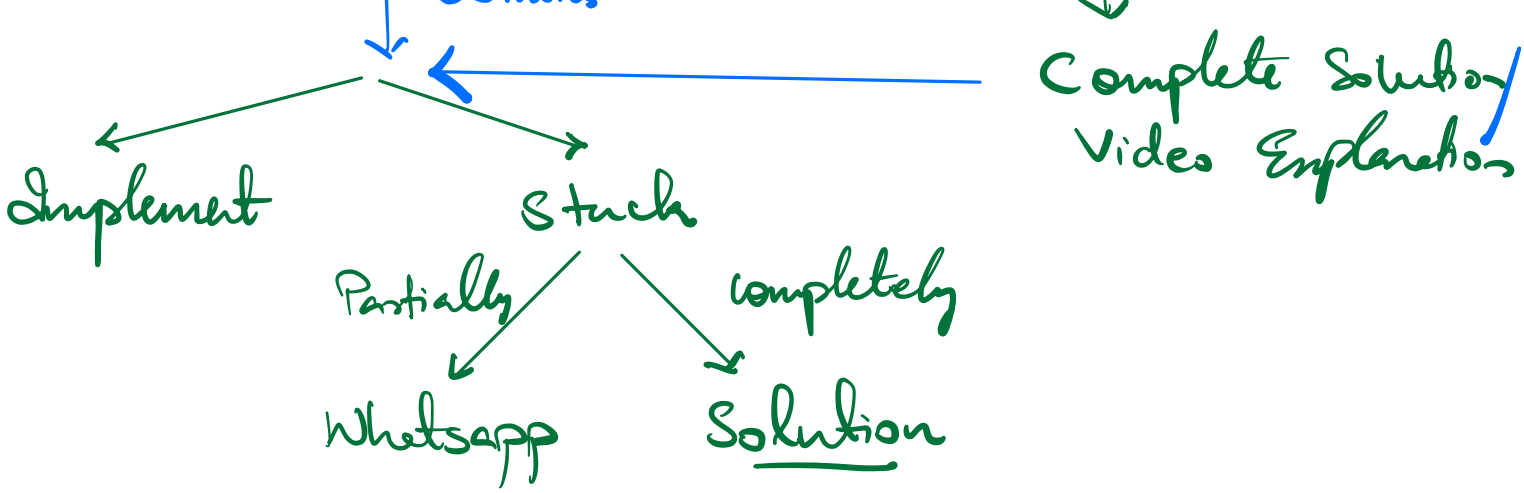
```
void selectionSort(A[], N) {

    end = N-1;
    for (j=0; j<N-1; j++) {   //(N-1) times
        maxEle = INT_MIN
        max_index = -1;
```

```
for (i = 0; i <= end; i++) {
    if (A[i] > maxEle) {
        maxEle = A[i];
        max_index = i;
    }
}
swap(A, max_index, end);
end--;
}
}
```

T.C. = $O(N^2)$
S.C. = $O(1)$




Complete Solution
Video Explanation
Implement
Stuck
Partially
completely
Whatsapp
Solution