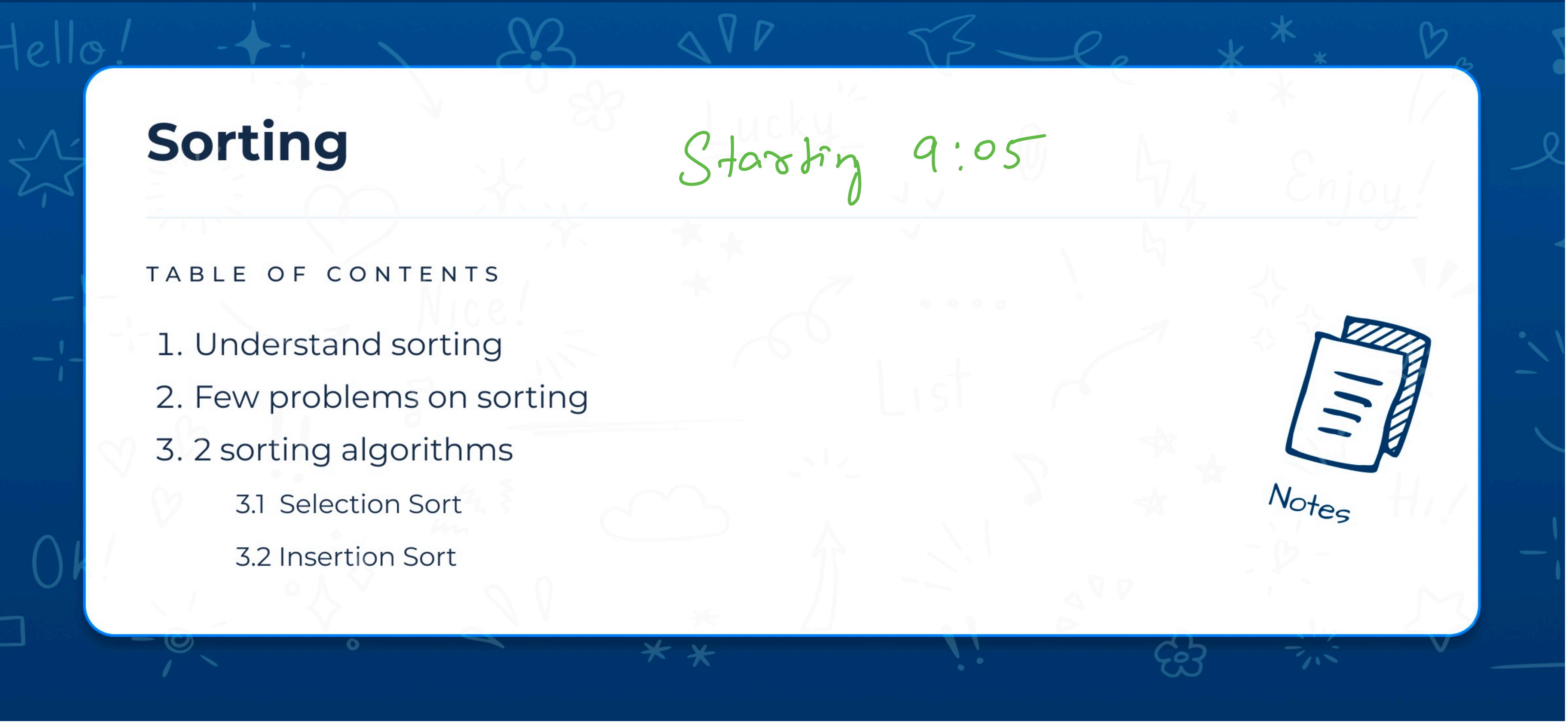


Sorting

Starting 9:05

TABLE OF CONTENTS

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

Notes

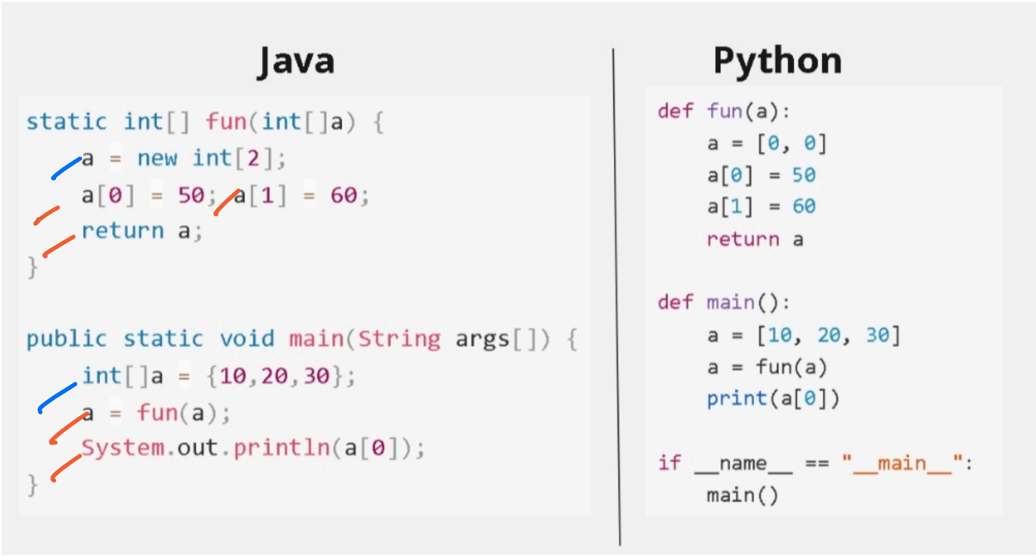
Revision Quizzes

1. Stacks / Heaps

2.

o/p - 50.

Predict the output :

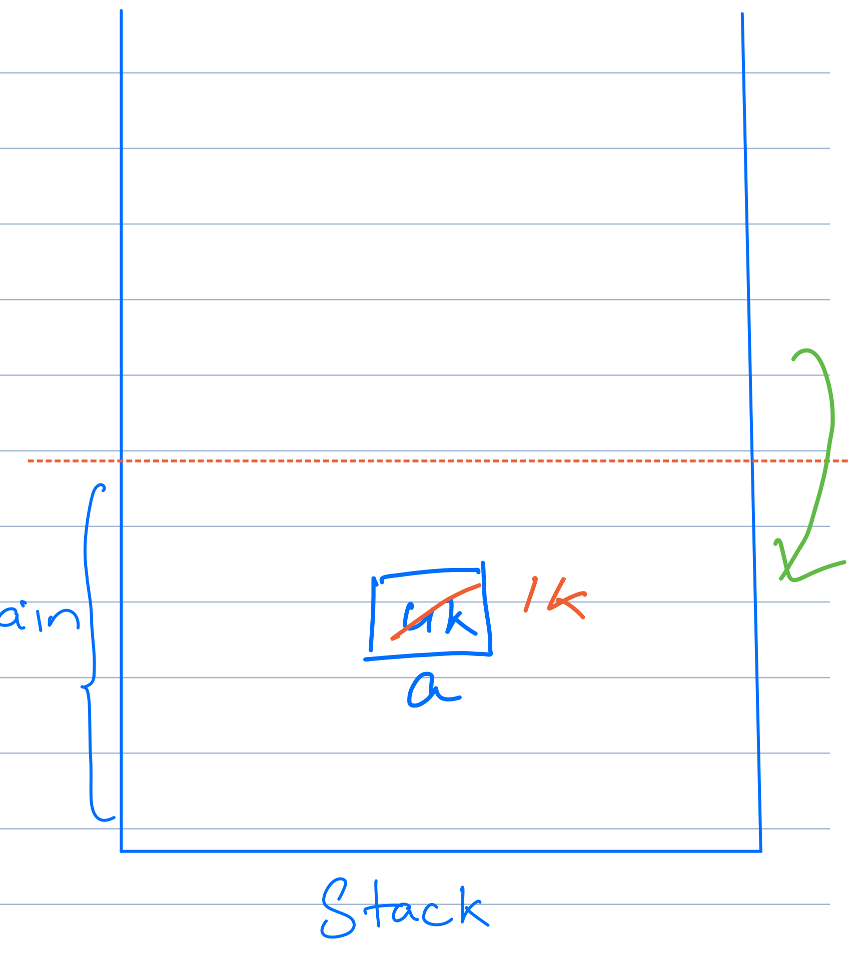


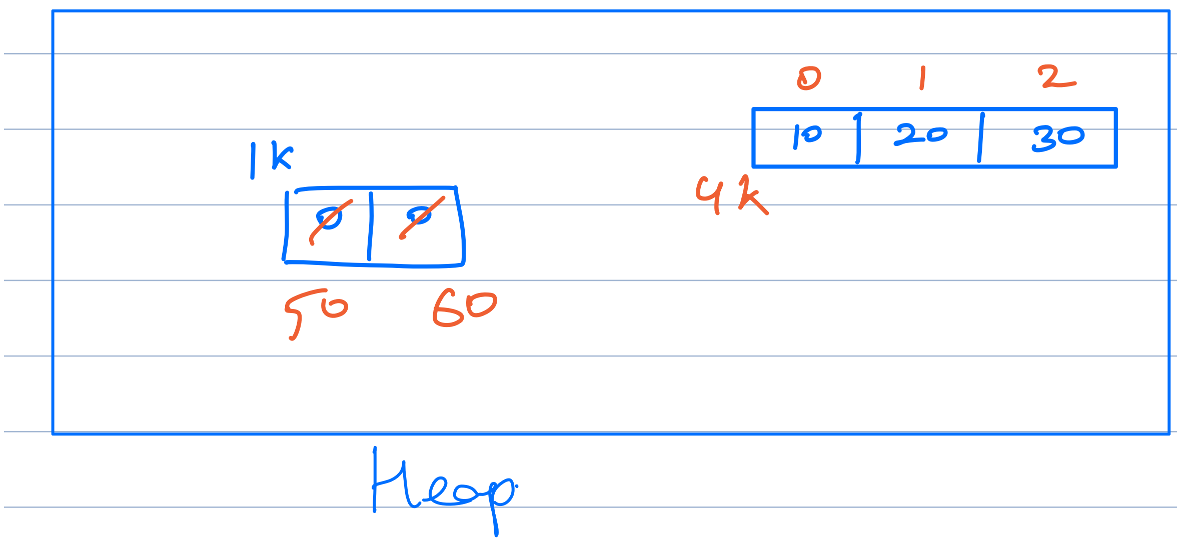
Java

```java
static int[] fun(int[]a) {
    a = new int[2];
    a[0] = 50; a[1] = 60;
    return a;
}

public static void main(String args[]) {
    int[]a = {10,20,30};
    a = fun(a);
    System.out.println(a[0]);
}
```

Python

```python
def fun(a):
    a = [0, 0]
    a[0] = 50
    a[1] = 60
    return a

def main():
    a = [10, 20, 30]
    a = fun(a)
    print(a[0])

if __name__ == "__main__":
    main()
```





Sorting :

An arrangement of data in a particular order based on some parameter.

ex:- 1. {2, 3, 9, 12, 17, 19}

Parameter:- Value ↑

2. {19, 6, 5, 2, -1, -19}

Parameter:- Value ↓

Quiz 1

	0	1	2	3	4
arr :-	1	13	9	6	12
	1	2	3	4	6.

Parameter:- Cout of factors ↑

Note:- In today's class, we will be using in-built libraries to sort data.

Java :-

Arrays.sort(arr) # Static arrays

Collections.sort(arr) # Array List

C++ :-

```
sort(arr, arr + N);          # Static arrays
sort(arr.begin(), arr.end());
                             # Vectors
```

Python:-

```
arr.sort();
```

T.C. → $O(N\log N)$

S.C → $O(N)$

Problem 1

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

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

Find minimum cost to remove all elements.

ex:- $\begin{array}{ccc} 0 & 1 & 2 \\ \{2, & 1, & 4\} \end{array}$

Ways to removal

a)

1    $1+2+4 = 7$

2    $2+4 = 6$

4   4

17

b)

2    $2+4+1 = 7$

4    $4+1 = 5$

1   1

13

c)

4    $1+2+4 = 7$

2    $1+2 = 3$

1   1

11 ↓ ans.

+ 3 more ;

1, 4, 2 → 15

2, 1, 4 → 16

4, 1, 2 → 12

Quiz 2:-

$$arr :- \{\overset{0}{4}, \overset{1}{6}, \overset{2}{1}\}$$

a)	1	1+4+6 = 11	b)	1	1+4+6=11	c)	4	1+4+6=11
	4	4+6 = 10		6	6+4 =10		1	1+6 =7
	6	6		4	4		6	6
		27			25			24

a)	4	1+4+6 = 11	b)	6	1+4+6=11	c)	6	1+4+6=11
	6	6+1 =7		1	1+4 =5		4	1+4 = 5
	1	1		4	4		1	1
		19			20			17

Generalising:-

ex:- $\{\overset{0}{a}, \overset{1}{b}, \overset{2}{c}, \overset{3}{d}\}$

Let's remove in the order	Cost
a	$a + b + c + d$
b	$b + c + d$
c	$c + d$
d	d
	$a + 2b + 3c + 4d$

$a > b > c > d$ to minimize

Solution Approach

Sort the array in decreasing order.

Iterate through the array & accumulate the value of arr[i] * (i+1) in the global sum..

Quiz 3

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

$\downarrow$ reverse - sort()

$\{\overset{0}{5}, \overset{1}{3}, \overset{2}{1}, \overset{3}{-3}\}$ i

ans += a[i] * (i+1)

$ans = \cancel{0}\ 5 + 6 + 3 + (-12) = \boxed{2}$

$2^{10} = 1024$

$\boxed{10^9}$

4aB.

Code

```
int   find Minimum Cost ( int [] arr , int N) {

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

T.C → O(NlogN)
S.C → O(N).

Problem 2

(distinct elements)

Given N array elements, calculate number of noble integers.

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

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

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

ans = 3.

0	1	2	3
-3	0	2	5

Q-4. { -3, 0, 2, 5 }   ans = 1.

Idea 1:- For every element, iterate over the entire array to find out the count of elements smaller than the current element.

T.C → $O(N^2)$

Idea 2:- Sort the array in increasing order.

Iterate through the array & increment ans if (arr[i] == i)

ex:-

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

Sorted arr :-

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

Code

```
NlogN { sort (arr);      int ans = 0;

      { for (i = 0; i < N; i++) {
N     {     if (arr[i] == i) {
      {             ans++;
      {     }
      }
         }

      return ans;
```

T.C → O(NlogN)

S.C → O(N).

Variation

What if the elements are not distinct ?

Q:- 5 arr:- {-10, 1, 1, 3, 100}

0	1	2	3	4
-10	1	1	3	100

ans = 3.

Q:- 6 arr :- {-10, 1, 1, 2, 4, 4, 4, 8, 10}

0	1	2	3	4	5	6	7	8
-10	1	1	2	4	4	4	8	10

ans = 5.

Q:- 7

arr:- {-3, 0, 2, 2, 5, 5, 5, 5, 8, 8, 10, 10, 10, 14}

0	1	2	3	4	5	6	7	8	9	10	11	12	13
-3	0	2	2	5	5	5	5	8	8	10	10	10	14

ans = 7.

Approach:- Sort the array & iterate on entire array. While iterating

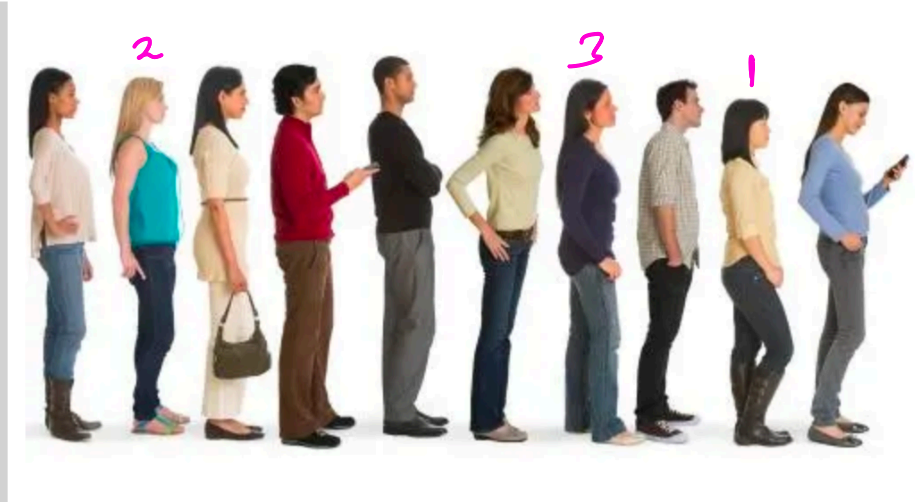
a) The curr. element is same as previous element :- The previous element is going to determine whether a[i] is noble or not.

b) The current element is different to previous element :- The index i of that element will determine whether a[i] is noble or not

Code → H.W.

10:25

Selection Sort



3

2.

1

Arranging Students Height-wise to form an assembly line.

ex:-

$$\begin{array}{cccc} 0 & 1 & 2 & 3 \\ \{\underset{2}{\cancel{5}}, & 6, & 4, & \underset{5}{\cancel{2}}\} \end{array}$$

$$\downarrow$$

$$\{2 \,\vdots\, \overset{4}{\cancel{6}}, \overset{6}{\cancel{4}}, 5\}.$$

$$\downarrow$$

$$\{2, 4, \,\vdots\, \overset{5}{\cancel{6}}, \overset{6}{\cancel{5}}\}$$

$$\downarrow$$

$$\{2, 4, 5 \,\vdots\, 6\}$$

Code

```
void  selectionSort ( int arr[], int N) {

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

0 1 2 3

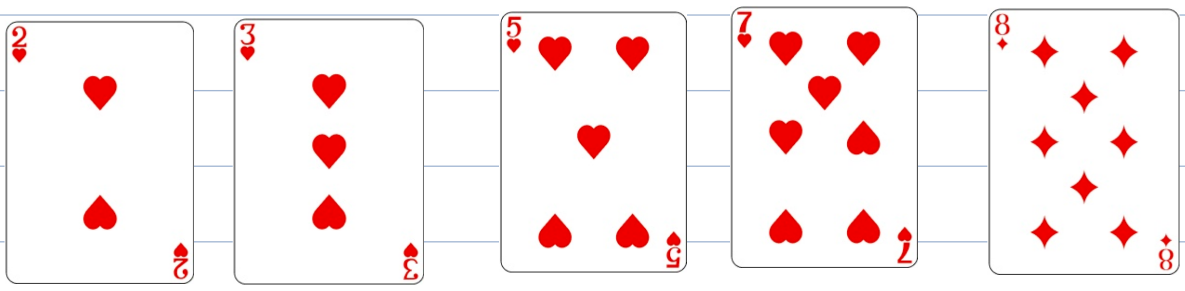
arr : { ~~5~~ 2 , ~~6~~ 4 , ~~4~~ ~~6~~ 5 , ~~2~~ ~~5~~ 6 }

i

T.C → $O(N^2)$

S.C → $O(1)$

Insertion Sort



ex:-

0	1	2	3	4
~~5~~	~~3~~	1	6	2
3	5			

↓

0	1	2	3	4
~~3~~	~~5~~ ~~1~~	~~1~~	6	2
1	3	5		

↓

0	1	2	3	4
1	3	5	6	2

↓

0	1	2	3	4
1	~~3~~	~~5~~	~~6~~	~~2~~ 6
	2	~~2~~ 3	~~2~~ 5	

↓

0	1	2	3	4
1	2	3	5	6

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; }.
    }
}
```

~~2~~ 3 i

0 1 2 3 4

ex:- { 1 , ~~3~~ 2 , ~~5~~ , ~~6~~ ~~2~~ , ~~2~~ 6 }.

j 1 5

↓

{1, 2, 3, 5, 6}.

T.C → $O(N^2)$

ex:- {5, 4, 3, 2, 1}.

i

ex:- {1, 2, 3, 4, 5}.

N-1

T.C → O(N) → Best Case.