

DAY  
58  
DSA

## Time & Space Complexity continued

Date :- 10 November, 2021

Day → Wednesday

- Agenda →
1. For 5 loop
  2. Polynomial
  3. Highest Frequency Character
  4. Insertion Sort.

Q1 → \*

हमें एक for loop दिया गया है। for loop, और उसे nested for loop की तरह (मात्र, तभी और कोई for loop नहीं) मात्र तभी तय for loop की time complexity  $O(n^2)$  होगी।

How?

For5.java

→ `for(unt i=1;j=1 ; i<=n ; j++) {`

// do some work

if(j>1)

{ j=1;

i++;

→  $n = 4 \rightarrow$  We need to dry run it for  $n = 4$

or you may take any other number.

for (int i = 1; j = 1; i <= n; j++)

{ // do some work - k

if (j > 1)

{  
j = 1;  
i++;

j

j

i	j	k	
1	1	k	k for i = 1
2	2	k	2k for i = 2
3	3	k	3k for i = 3
3	4	k	4k for i = 4
4	2	k	
4	3	k	
4	4	k	
5	1	k	ignor

$$\rightarrow k + 2k + 3k + 4k$$

$$\rightarrow k(1+2+3+4)$$

$$\rightarrow k \left[ \frac{n(n+1)}{2} \right] \text{ AP}$$

$$= k \left[ \frac{(4)(5)}{2} \right]$$

$$= k \left[ \frac{20}{2} \right]$$

$$= 10k$$

# DRY RUN FOR $n=4$

classmate

Date \_\_\_\_\_  
Page \_\_\_\_\_

$i$

$\rightarrow 1$

$i = 1$  and  $j = 1$  initially जिससे  
रक्कम 'K' हुआ

अब condition check करें तो

$j$  is not greater than  $i$  तो वापस loop  
उपर गया तो अब  $j$  की value बढ़ कर  
2 हो गई

$j$

$\rightarrow 1$

for (init  $i=1, j=1; i \leq n; j++$ )

{

// do some work

if ( $j > i$ )

{

$j = 1$

$j++$

3

$21$

$\rightarrow 12$

अब condition check करी  $j > i$  की  
रक्कम 'K', और कि  $i=2$  करदौ और  $j=1$  as per the condition

तो for  $i=1$ , loop केवल 1 बार चला

When  $[i=1, j=1]$ , work = K क्योंकि second time loop  
complete होने से पहले ही  
 $i$  की value change हो गई

\* So, now, we have  $i=2, j=1$ , condition check की,  $i \leq n$   
 $i$  अभी 2 है तो valid हो loop के अंदर जाना जो loop के अंदर  
आ गया, अब रक्कम 'K', now  $j$  is not greater than  $i$   
तो loop से बाहर आ जाए, अब  $j$  की value बढ़ कर 2 हो जाएगी  
जब हम ऊपर जाएगा

$\rightarrow 2$

$2$

अब loop चला क्योंकि  $i=2$  &  $i \leq n$ , तो हम loop के अंदर आ जैए  
, अब रक्कम 'K', अब check की condition वापस से, जो  $j$  is not  
greater than  $i$  if loop से बाहर आ जाए and value of  $j$   
increases

i

j

3 1

$\rightarrow 23$

अब loop की condition check होगी कि  
is  $i \leq n$ ,  $2 \leq 4$ , तो इस loop के  
अंदर आ जायेगा 3rd condition.  
check होगी कि is  $j > i$ ,  $3 > 2$  तो अब  
 $i=3$  and  $j=1$  हो जायेगा as per the  
conditions.

तो for  $i=2$ , loop के कल 2 वाले कर्ता

first  $\rightarrow i=2, j=1$  work=k

second  $\rightarrow i=2, j=2$  work=k

\* So, now  $i=3, j=1$  तो अब loop की condition check की if ( $i \leq n$ )  
 $3 \leq 4$ , तो अब कम हुआ 'k', अब check की condition वापस से,  
कि is  $j > i$  नहीं  $j$  is not greater than  $i$  ( $1 < 3$ ) तो अब  
loop से बाहर आ गए, और अब उपर वापस आए तो  $j$  की value 2 कर  
2 हो गई

3

2

अब हमने loop की condition check की, is ( $i \leq n$ ), yes  $3 \leq 4$ , तो कम हुआ  
'k', अब second condition check होगी is ( $j > i$ ),  $2$  is not greater than  
 $3$  तो loop exits and  $j$  increases जब इस अंदर आए से तो, तो now  
 $j=3$  and  $i=3$

3

3

तो अब loop की first condition check की, is  $i \leq n$ ,  $3 \leq 4$ , तो कम हुआ 'k'  
3rd second condition check होगी is ( $j > i$ ),  $3$  is not greater than  $3$  तो  
loop exits और  $j$  increases. जब इस अंदर आए से तो  $j$  increases  
हो जाएगा और  $j$  की value हो पाएगी 4

for (int i=1; j=1; classmate

; i<=n; j++)

Date \_\_\_\_\_  
Page \_\_\_\_\_

I do some work.

if ( $j > i$ )

{  
 $j = 1;$

$i++;$

j

j

J J  
34 41

तो अब loop की condition check की (is  $i \leq n$ )  
 yes ( $i < 4$ ) तो loop के अंदर 3 पर J का,  
 work done is 'k', और second condition  
 check की, is ( $j > i$ ) तो yes  $4 > 3$  तो  
 now i will increase to 4 and j  
 will decrease to 1

\* तो for (i=3, loop 3 तक)

for (int i=1, j=1; i<=n; j++) {  
 } // do some work

if ( $j > i$ )

{  $j = 1$  ;

i++;

}

first       $i=3, j=1$

work done = k

second     $i=3, j=2$

work done = k

third      $i=3, j=3$

work done = k.

\* So now,  $i = 4, j = 1$ , loop की condition check की is ( $i \leq n$ )  
 is  $4 \leq 4$ , yes तो loop के अंदर 3 पर J का, काम किया 'k', और  
 j is not greater than i, ऐसा पहला when we checked second  
 condition (is ( $1 > 4$ )) no तो इस loop के बाहर आया  
 और अब ऊपर गए तो j की value 0 से हो गयी 2

4 2

तो अब इस loop की condition check की है, is ( $i \leq n$ ) ( $4 \leq 4$ ). इस  
 loop के अंदर 3 पर J का काम किया 'k', और loop की condition check की  
 is ( $j > i$ ) 2 is not greater than 4 तो loop के बाहर 3 पर J की value 3 हो गयी।

4 3

तो अब  $i=4$  और  $j=3$  हैं तो अब loop की condition check हो, जैसे  $i \leq n$ , Yes  $4 \leq 4$  तो loop के अंदर आ JRD तो अब second condition check होगी।

classmate

Date \_\_\_\_\_

Page \_\_\_\_\_

जैसे ( $j > i$ ) तो  $3$  is not greater than  $4$  तो loop से बाहर आ JRD और अब work करना होगा तो कहा कि  $j$  will increase to  $4$  now.

$i$

$j$

$4$

तो हमने अब जब loop की condition check की है तो ( $i \leq n$ ) ( $4 \leq 4$ ) तो loop के अंदर आ JRD तो work check किया जैसे ( $j > i$ )  $4$  is not greater than  $4$  तो loop से बाहर आ JRD और JRD के बाद  $j$  की value increase हो JRD so now  $j$  will be 5.

for (int  $i=1$ ,  $j=1$ ;  $i \leq n$ ;  
 $j++$ )

{

//do some work  
if ( $j > i$ )

{

$j=i$ ;  
 $i++$ ,

}

$j$

# 45 51

So now check the loop condition,  $is(i \leq n)$  yes  $4 \leq 4$ , फिर किया 'K', then second condition check हो जैसे  $j > i$  is ( $5 > 4$ ) → Yes, तो loop के अंदर आ JRD तो  $j$  will decrease to 1 and  $i$  will increase to 5.

So now  $i=5$ ,  $j=1$  यह तो  $j$  नहीं उसी तरह होगा for

$i=5$  &  $j=1$

∴ 4<sup>th</sup> for  $i=4$ , loop 4 JRD चला

first  $i=4$ ,  $j=1$

work done = k

second  $i=4$ ,  $j=2$

work done = k

third  $i=4$ ,  $j=3$

work done = k

fourth  $i=4$ ,  $j=4$

work done = k

## Time Complexity

$$n=4 \rightarrow T(n) = K + 2K + 3K + 4K$$

for n.  $T(n) = (K + 2K + 3K + 4K + \dots + nK)$

$$T(n) = K(1 + 2 + 3 + 4 + \dots + n).$$

$$T(n) = K(n)(n+1)$$

$$= K \frac{n^2}{2} + Kn$$

$$T(n) \propto n^2$$

∴ Time Complexity =  $O(n^2)$

Ques. 2

## CALCULATE POLYNOMIAL

दर्शाएँ x और n की value दिए गए होगी और हमें polynomial की value निकालनी होगी using the formula

$$\rightarrow \boxed{\text{Polynomial} = 1 \cdot x^n + 2 \cdot x^{n-1} + 3 \cdot x^{n-2} + \dots + n \cdot x^1}$$

तो हमें पैसे function (प्रकल्प दोगा जो हमें polynomial की value निकालके देंगा)

$$\text{example} \rightarrow x = 10 \\ n = 3$$

$$\text{Polynomial}(x, n) \rightarrow \text{Polynomial}(10, 3)$$

$$\begin{aligned} \text{Polynomial} &= 1 \cdot x^3 + 2 \cdot x^2 + 3 \cdot x^1 \\ &= 1 \cdot 10^3 + 2 \cdot 10^2 + 3 \cdot 10^1 \\ &= 1 \cdot 1000 + 2 \cdot 100 + 3 \cdot 10 \\ &= 1000 + 200 + 30 \\ &= 1230 \end{aligned}$$

में इस question को  $O(n)$  में ही solve करना है।

पहले function # loop को reverse (उलटा) करना पड़ेगा।

$$\text{Polynomial} = 1 \cdot x^n + 2 \cdot x^{n-1} + 3 \cdot x^{n-2} + \dots + n \cdot x^1$$

$$\rightarrow 1 \cdot x^n + 2 \cdot x^{n-1} + 3 \cdot x^{n-2} + \dots + (n-3) \cdot x^4 + (n-2) \cdot x^3 + (n-1) \cdot x^2 + n \cdot x^1$$

Loop को उलटा कराएं

$$\rightarrow mx^1 + (n-1)x^2 + (n-2)x^3 + \dots + (2)x^{n-1} + (1)x^n$$

Coefficient      Power of  $x$

$$c = \text{coefficient}$$
$$box = x.$$
$$a = 0$$

$$\left. \begin{array}{l} \text{coefficient} \\ \text{power of } x \\ \text{ans} \end{array} \right\}$$

if Code  
Snippet

(code  
कोड  
स्क्रिप्ट  
टिप्पणी)

while ( $c >= 1$ )

{

    term =  $c * box;$   
    ans = ans + term  
     $c--;$   
     $box = box * x;$

# CALCULATE POLYNOMIAL <Code>

```
→ import java.util.*;
→ import java.io.*;
```

```
→ public class Main
```

```
{
```

```
→ public static void main (String [] args)
```

```
{
```

```
→ Scanner scan = new Scanner (System.in);
→ int x= scan.nextInt();
→ int n = scan.nextInt();
```

```
→ int c = n;
```

// coefficient

```
→ int pox = x;
```

// power of x

```
→ int ans = 0;
```

// answer

```
→ while (c >= 1)
```

```
{
```

```
→ int term = c * pox;
```

```
→ ans = ans + term;
```

```
c--;
→ pox = pox * x;
```

```
}
```

```
→ System.out.println (ans);
```

```
}
```

# DRY RUN THE CODE

Date \_\_\_\_\_  
Page \_\_\_\_\_

- \* हमने coefficient को रखा  $n$   
के बराबर
- \* Power of  $x$  को रखा  $x$  के  
बराबर
- \* answer को रखा 0
- \* Code तक तक चलेगा जब तक  
coefficient 1 या उससे छोटा हो।  
मतलब  $1 \cdot x^n$  तक चलेगा।
- \* term होगा हमारा  $c * pow$

i.e.  $n \cdot x^i$        $i$  → Power  
 Coefficient       $x^i$

```

int c = n;
int pox = x;
int ans = 0;
while (c >= 1)
{
    int term = c * pox;
    ans = ans + term;
    c--;
    pox = pox * x;
}
System.out.println(ans);
    
```

- \* बाद में ans में term को add करते चलो।
- \* end में coefficient किया decrease करोकि हमें coefficient  
को  $n, n-1, n-2, \dots, 1$  तक ले के जाना है।  
So we need to decrease  $c$ , so that at the end it can move  
till 1.
- \* end में pox को किया  $pox * x$  करकि  $x, x^2, \dots, x^n$  हो।  
मतलब cmd में।
- \* In the end, just print the answer.

Ques. 3.

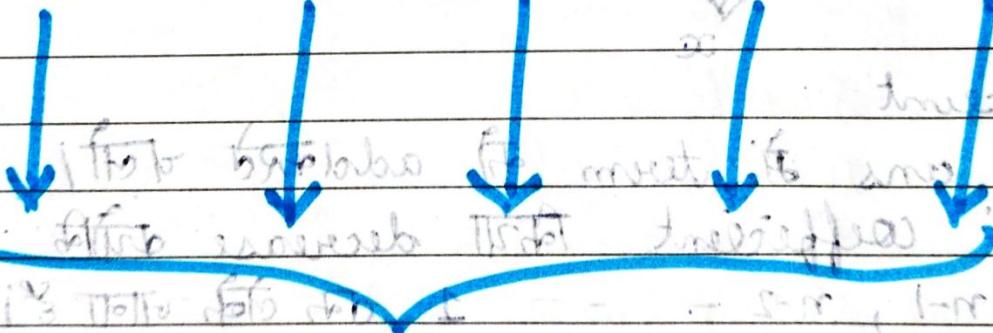
# HIGHEST FREQUENCY CHARACTER

classmate  
Date \_\_\_\_\_  
Page \_\_\_\_\_

→ मेरे एक string दिया गया है, उसी string को character क्रेंट करना है (विस्तार से frequency सबसे ज्याका होगा)

- \* So basically you are given a string, let's say "abrgcadabca" you have to print the character that has the highest frequency. ie 'a' = whose frequency is 5

a b r a c a d a b r a



FREQUENCY OF A = 5

- \* Time Complexity should be  $O(n)$  (Highest)
- \* Input given  $\rightarrow$  a string
- \* Output  $\rightarrow$  Character with higher frequency.

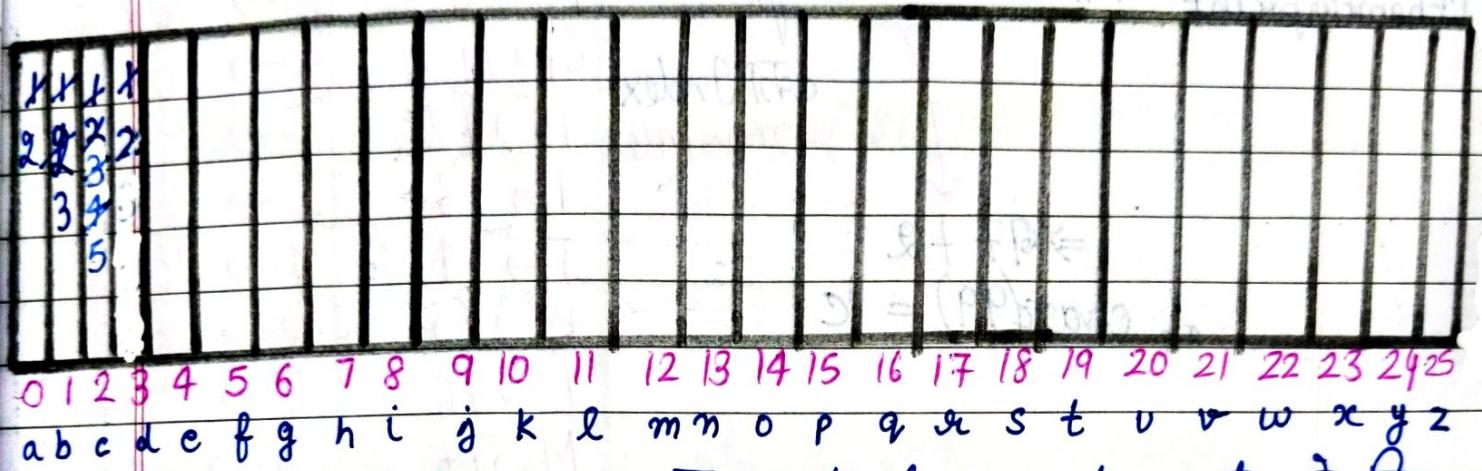
## Basic Idea

classmate

Date \_\_\_\_\_  
Page \_\_\_\_\_

- \* इसमें हम एक 26 size का array बनाएँगे, और हर spot को स्कॉप particular character के लिए fix रखते हैं और हम अभी के लिए assume करते हैं कि string में क्या small characters हो आएँगे,

- \* Let's say the string is  $bcddabccac$



हर एक spot array को स्कॉप particular character के लिए fix किया गया है।

- \* तो अब string में traverse करते हैं और जो भी character आता जाएगा, उसकी frequency बढ़ाते जाएँगे

Similarly  $bcddabbbccac$  में हमें c 5 बार मिला तो ये होगी highest frequency.

'c'  $\rightarrow$  5 times है frequency, तो ये हमें book काना पड़ेगा ये check करने के लिए किसी से बड़े index पर array के Maximum value है तो we got 5 is the maximum value

- \* We got the highest frequency as '5'

लेकिन हमें highest frequency नहीं पाइए / Instead हमें ये character का जिसकी frequency highest है / so, we want 'C'

\* तो अब 'c' का index चाहिए तो subtract  
 the ascii of 'a' from ascii of 'c', द्यें c के  
 index के maxIndex लोग

$$'c' - 'a' = 99 - 97$$

maxIndex = 2.

\* तो अब हमें 'c' का index मिल ही  
 गया है तो अब हमें [a + maxIndex]

ASCII  
 का character point करना

$$\Rightarrow 97 + 2$$

$\therefore \text{char}(99) = 'c'$

contIndex  
 = maxIndex

ASCII	
A → 65	a → 97
B → 66	b → 98
C → 67	c → 99
D → 68	d → 100
E → 69	e → 101
F → 70	f → 102
G → 71	g → 103
H → 72	h → 104
I → 73	i → 105
J → 74	j → 106
K → 75	k → 107
L → 76	l → 108
M → 77	m → 109
N → 78	n → 110
O → 79	o → 111
P → 80	p → 112
Q → 81	q → 113
R → 82	r → 114
S → 83	s → 115
T → 84	t → 116
U → 85	u → 117
V → 86	v → 118
W → 87	w → 119
X → 88	x → 120
Y → 89	y → 121
Z → 90	z → 122

# HIGHEST FREQUENCY CHARACTER <code>

classmate

Date \_\_\_\_\_

Page \_\_\_\_\_

```
→ import java.util.*;  
→ public class Main {  
→     public static void main (String [] args) {  
→         Scanner scan = new Scanner (System.in);  
→         String str = scan.nextInt();  
→         int freq [] = new int [26]; // इसका freq के  
→         ↴ frequency  
→         array          0th Index QR  
→         a की frequency,  
→         1st idx QR b की freq  
→         इत्यादि पाएंगे & so on...  
→         for (int i=0; i< str.length(); i++) // string QR  
→             ↴  
→             ↴ char ch = str.charAt (i); } string के ith  
→             int idx = ch - 'a'; } under तोला  
→             freq [idx]++; } character की संख्या  
→             ↴ उस particular string  
→             के alphabet की जिस  
→             index पर store करना है  
→             वो index निकालने के लिए  
→             subtract करके character से 'a'.  
→             eg → 'c' - 'a' = 99 - 97 = 2.  
→             तो c का idx वॉले 2 है।  
→             → अब मैं maxidx को 0 सार्कारौ हूँ।  
→             int maxidx = 0;  
→             for (int i=1; i< freq.length; i++)  
→                 ↴  
→                 ↴ if (freq [i] > freq [maxidx]) { 3) 3) 3)  
→                     ↴ maxidx = i;  
→                     ↴ then update  
→                     maxidx to i  
→                     so now maxidx = 1.
```

→ `char mfc = (char)(maxIndex + b');`  
→ `System.out.println(mfc);`

} } }

maxindex  $\Rightarrow$  'a' - 97

add 48  $\Rightarrow$  97

maxindex

char

of ASCII code

Get UTT /

48 97 ASCII code

(char) it type cast

48 97 it

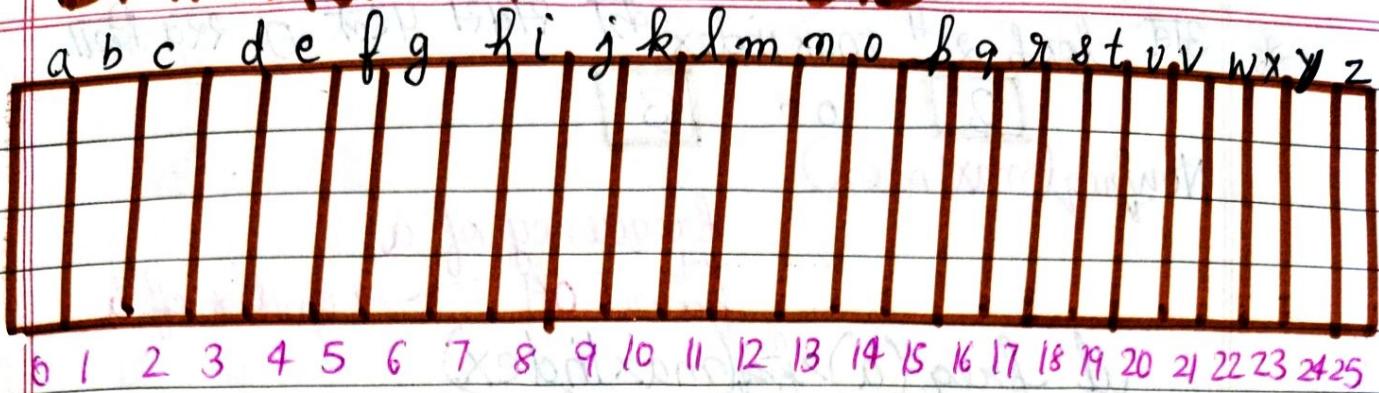
maxindex of it

character it

UTT /

# DRY RUN THE CODE

Date \_\_\_\_\_  
Page \_\_\_\_\_



String → aa bbb cbc ccc dd ab b

X	X	X	X
X	X	X	2
3	3	3	
4	4	4	
5			
6			

\* सबसे पहले Integer array में '0' stored है वर्ष Index 0, फिर सारे process होंगे, जिसे a' मिला तो उसकी frequency बढ़ जाएगी 0 to 1, next item फिरसे अपर अपर a' मिला

then a frequency जो भाईजी from 1 to 2.

इसी ही string के सभी character को traverse और जो उसी character मिलता जाएगा, उसी character के corresponding Index के frequency बढ़ावी जाएगी।

```

→ int freq = new int[26];
→ for (int i = 0; i < str.length();
    loop
    {
        char ch = str.charAt(i);
        int idx = ch - 'a';
        freq[idx]++;
    }
}

```

```

→ int maxIndex = 0;
→ for (int i = 1; i < freq.length(); i++)
    if (freq[i] > freq[maxIndex])
        loop
        {
            maxIndex++;
        }
}

```

```

→ char mfc = (char)(maxIndex + 'a');
→ System.out.println(mfc);
}

```

a → 3  
b → 6  
c → 5  
d → 2.

Now, for Loop 2

\* अब loop 2 का max index को सबसे पहले 0, 22वां प्रिया

3 or 6

$\text{freq}(\text{maxIndex})$

frequency of b

$\text{arr}[1]$ , Index of b

if  $\text{freq}(b) > \text{freq}(\text{maxIndex})$

{ then,  $\text{maxIndex}$  = Index of b.

$\text{maxIndex} = 1$ .

\*

6

or

5

$\text{freq}(\text{maxIndex})$  frequency of c

$\text{arr}[2]$ , Index of c.

if  $\text{freq}(c) > \text{freq}(\text{maxIndex})$

{ then update maxIndex to index of c }

$5 < 6$

$\text{freq}(c)$   $\text{freq}(\text{maxIndex})$

So, maxIndex will be the same - i.e 1

\*

6

or

2.

$\text{freq}(\text{maxIndex})$  frequency of id.

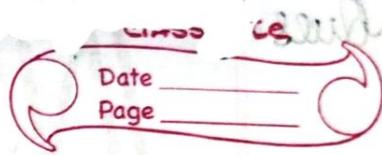
$\text{arr}[3]$ , Index of d.

if  $\text{freq}(d) > \text{freq}(\text{maxIndex})$

{ then update maxIndex to index of d }

So, maxIndex will be the same - i.e 1

So, now the loop ends, & the  
maxidx will get us 1.



अब हमें जो maxidx 1 में से इसके 'b' के से क्या होगा?

'a' + 1

↓      ↓

97 + 1

↓      ↓

98 = ASCII Code.

→ Now the ASCII code of character is 98.

To convert the ASCII code to character.  
we need to do explicit type casting

(char)(98) = 'c'

↓

So, we did explicit type casting like this

# INSERTION SORT

Insertion Sort is the sorting mechanism where the sorted array is built having one item at a time. The array elements are compared with each other sequentially and then arranged simultaneously in some particular order. The analogy can be understood from the style we arrange in a deck of cards.

This sort works on the principle of inserting an element at a particular position, hence the name Insertion Sort.

5	1	6	2	4	3
---	---	---	---	---	---

हम इस Array को  
अपना Input Set कहते  
चलते हैं।

Insertion Sort में  
हम key को pick  
करते हैं और उस  
key की उसके पहले  
वाले ट्रम्पी elements के  
साथ compare करते हैं  
और key की उसकी  
correct position पर  
रखते हैं।

5	1	6	2	4	3
1	5	6	2	4	3
1	5	6	2	4	3

1	2	5	6	4	3
1	2	4	5	6	3

1	2	3	4	5	6
---	---	---	---	---	---

→ 5 has nothing before it

→ 1 की 5 से compare  
करते हैं और 5 से

पहले insert करते हैं  
because 1 is  
smaller than 5

→ 6 is greater than 5  
& 1

→ 2 is smaller than 6 &  
5 but greater than 1,  
so it is inserted after  
1.

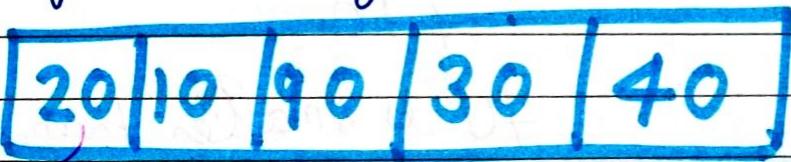
→ Some process happens with 3

And this process goes with all the elements if size of array is more.

- \* **Insertion Sort** स्टेप्स Comparison based sorting technique  
एवं एक array को sort करती है द्वारा shifting the element one by one from an unsorted sub-array to the sorted subarray.
- \* पहली इसके iteration होती है; तो हमें पहला element (unsorted) Input में से किया जाता होता है और वह us inserted in the sorted list at the correct location.

**Step 1)** → Always remembers, Array of 1 size is always sorted". If there is just one element, there is no other member to compare with hence it is always sorted (both increasing & decreasing)

→ So, if the array is of 5 elements



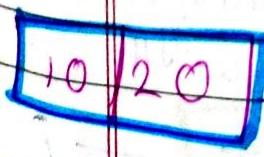
So, we assume that  
the element at first Index is  
already sorted.



Step 2) यही element है, So array of 1 size will always  
Journey 1 sorted.



3rd we need to decide the correct position of 10. Because 20 is greater than 10, so 10 will be placed before 20. So, the final order will be 10, 20.



Step 3

10	20	90
----	----	----

So, now we need

to decide the

correct position of 90.

classmate

Date \_\_\_\_\_

Page \_\_\_\_\_

Journey 2

10	20	90
----	----	----

90 is greater than 10 & 20.

So, it is at its correct position already.

Step 4)

10	20	90	30
----	----	----	----

So, we need to decide  
the correct position of

30.

Journey 3

10	20	30	90
----	----	----	----

90 is greater than 30, 10, 20.  
30 is smaller than 90.

Step 5)

10	20	30	90	40
----	----	----	----	----

So, now we need to  
decide correct position  
of 40.

40 is smaller than 90.

10	20	30	10	90
----	----	----	----	----

SORTED

INSERTION SORT में

REVERSE BUBBLING  
TECHNIQUE USE होती है।

3	4	2
---	---	---

Step 1) पहले 4 और 2 compare होते क्योंकि 4 से

immediate 2 compare होते हैं वो साथ में हो

3	2	4
---	---	---

2) फिर 3 और 2 compare होता है।

2	3	4
---	---	---

2	3	4
---	---	---

# INSERTION SORT <code>

classmate

Date \_\_\_\_\_

Page \_\_\_\_\_

```
→ import java.util.*;  
→ public class Main {  
    public static void main (String [] args) {  
        Scanner scan = new Scanner (System.in);  
        int n = scan.nextInt(); → n किया है Input (size of array)  
        int [] arr = new int [n]; → array बनाया array.  
        for (int i=0; i<arr.length; i++) {  
            arr[i] = scan.nextInt(); } array में input करते  
        }  
        for (int i=1; i<=arr.length-1; i++) {  
            for (int j=i-1; j>=0; j--) {  
                if (arr[j] > arr[j+1]) {  
                    int temp = arr[j];  
                    arr[j] = arr[j+1];  
                    arr[j+1] = temp; } }  
            }  
        else { break; }  
    }  
}
```

journey दौरा पर  
sort होता है  
सब elements  
पर compare  
होती है तो  
Reverse Order  
PTR (j+1) next element  
ये छापा होता है (j) position  
of current element  
then stop the loop there be break.

```
for (int val: arr){
```

```
    System.out.print(val + " ");
```

```
}
```

```
System.out.println();
```

```
}
```

```
}
```

# DRY RUN THE CODE

20 10 90 40 15  
0 1 2 3 4

\* Journey की तरीफ़ ? → Length of Array - 1  
→ 5 - 1  
→ 4

## Journey 1

20 \* 10 90 40 15  
0 1 2 3 4

SORTED

0th Index वाला element नो sorted ही है, अब focus होगा 1st Index पर कि 3 स्टेपों sort करके कैसे लिया करें insert करा दूँड़े।

## Journey 1 Result

20 \* 10 90 40 15  
0 1 2 3 4

10 20 \* 90 40 15  
0 1 2 3 4

## Journey 2

10 20 \* 90 40 15  
0 1 2 3 4

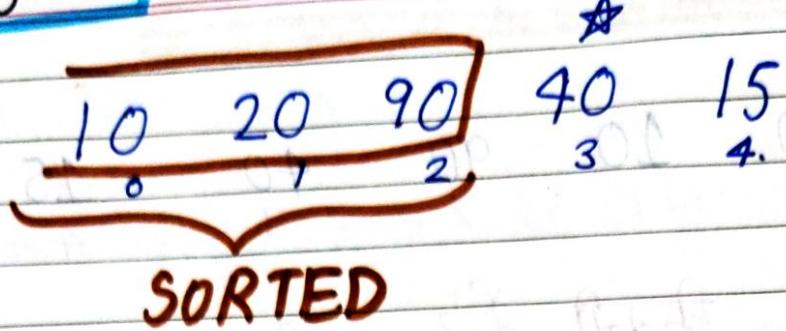
SORTED

1st Index. तक हम sorted मानेंगे, अब 90 को इस तरीके से insert करेंगे जो sorted रहेगा

## Journey 2 Result

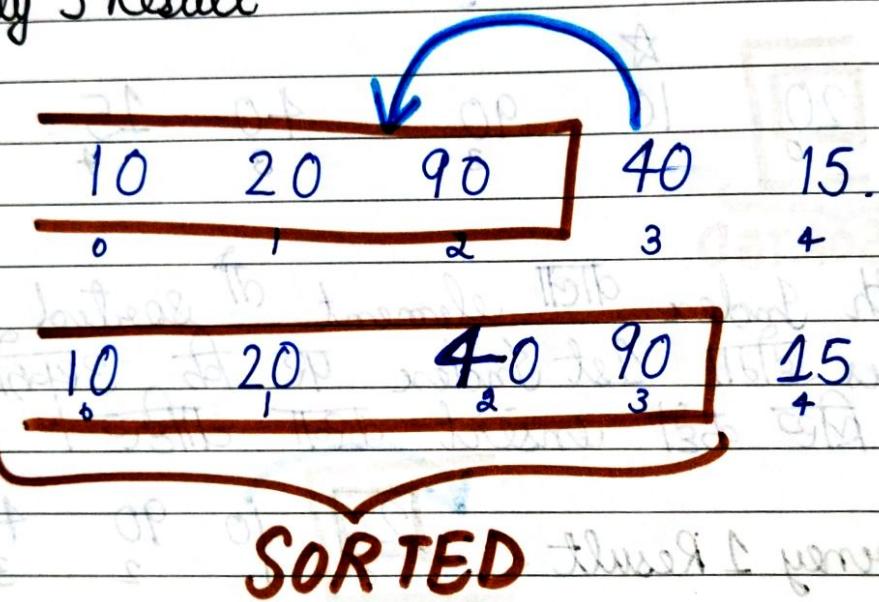
10 20 90 40 15  
0 1 2 3 4

### Journey 3

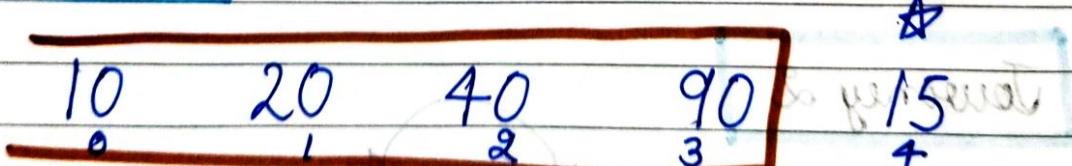


2nd Index ~~is~~ sorted  $\therefore$ , it focus on 3rd Index  
~~it~~ so as to insert it in such a way that  
array with 3rd Index is sorted.

### Journey 3 Result

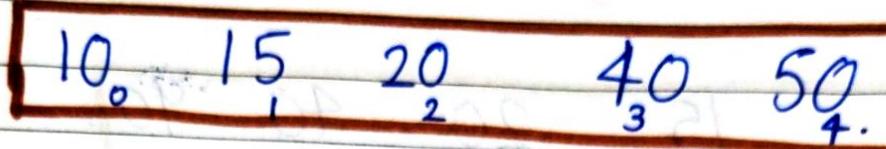
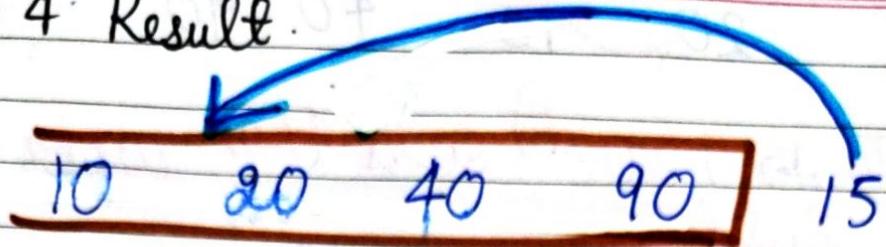


### Journey 4



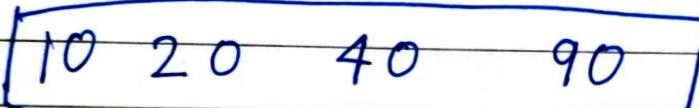
3rd Index ~~is~~ already sorted  $\therefore$   
focus on the 4th Index ~~so that~~ that after  
insertion the array still remains  
sorted.

## Journey 4 Result



### \* Journey 4 details

अब हम focus करेंगे on the Inner loop & we expect that the outer loop works fine, this is called layering.



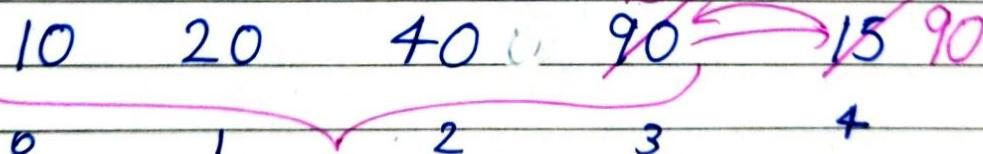
(15)

ये एक sorted array है।

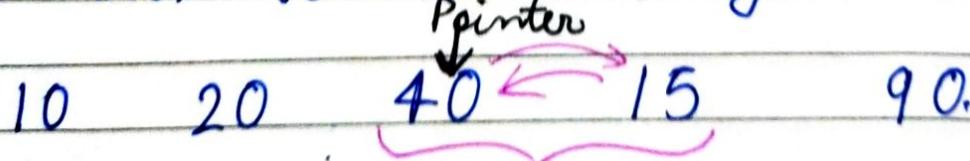
ईसी sorted array  
में हमें इसका नया  
element को insert

Pointer  
↓  
15

करना है और यह  
element है 15.



हमने मान लिया कि Index 3 तक सब sorted हैं, अब  
हम करेंगे Reverse Bubbling.



→ अब 40 और 15 को sort किया जाएगा using Reverse Bubbling

Pointer



10

20

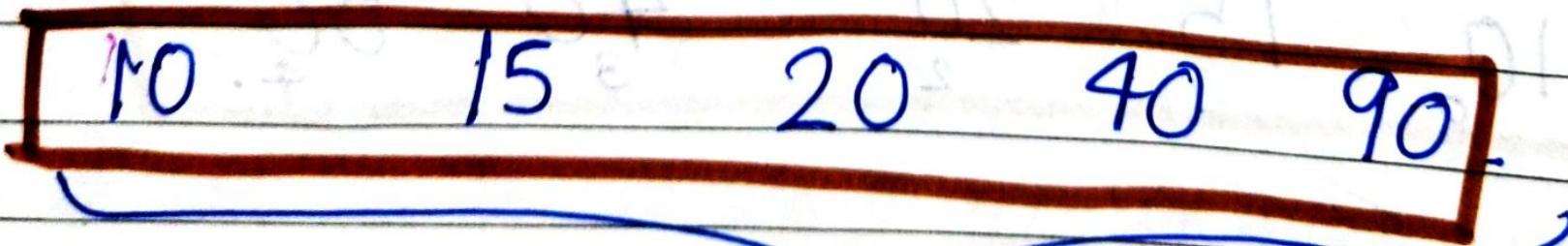
15

40

90

At 15, 20 of sort first  
Bubble sort.

using Reverse



Whole Array is Sorted → Final Result