

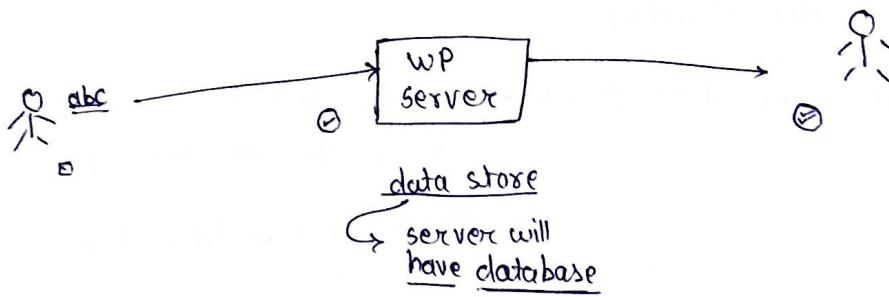
Data StructureData - Setin Java notes

Organization of data → Data structure

Digital data - audio, video, text, image

raw data → information → knowledge → decision making

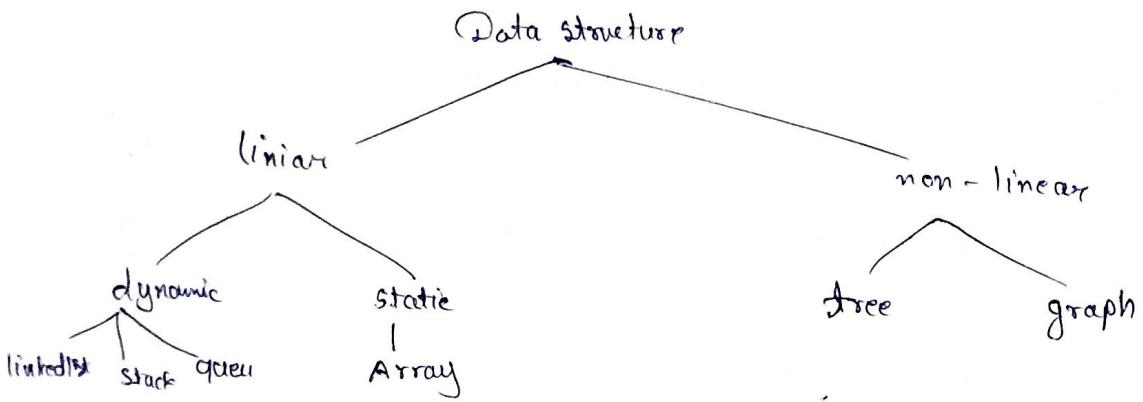
DBMS - Data base management system.

Data structure + DBMS ⇒ Backendeasy organization & storage  
of data.Data Base ⇒ collection of datae.g. gmail  
whatsappMobile database ⇒ 32 GB or 64 GB → Backup data storesecondary  
memoryActive memory ⇒ 2 GB or 4 GB → applications run  
in RAMPrimary  
memory

in Computer - Hard disk - secondary - external hD - Pendrive

RAM - primary memory

operation → insertion  
deletion  
updation  
selection



Array - contiguous memory allocation

int type one element - 4 byte      32 bit compiler  
8 bytes      64 bit compiler

Array - static linear data structure.

purely static linear data structure - मतलब size runtime के ही पता हो जानी चाहिए ,

But there are some tricks for smart programming .

int a[] = new int[5];

for giving value - we use loop

always start indexing with 0.

$i < a.length$

or  $i \leq a.length - 1$

$a.length = 5$ , if  $\text{int } a[J] = \text{new int}[5]$ ;

use two loops - 1 for loop for taking values of array elements  
2 for loop for taking displaying values.

optional, similar as in C

`int a[5];  
a = new int[5];`

→ array बनाना दो line में

in C

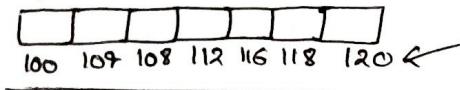
`int a[5];`

array बनाएगा 5 size की

Array - collection of similar type of data elements.

index → 0 1 2 3 4 5 6

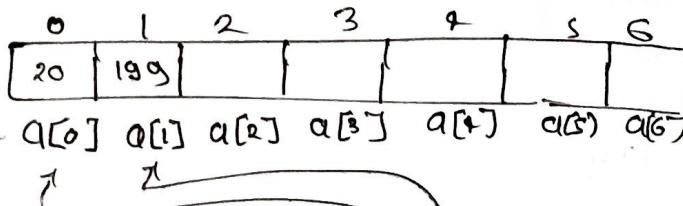
a



memory allocation

values → 60 90 100 40 20 10 50

name of  
memory  
allocation.



$a[0] = 20;$

$a[1] = 199;$

s.o.p (a[1]); → 199

s.o.p (a[0]); → 20

- Programs -
1. Array length
  2. sum of array elements
  3. multiply of array elements
  4. Searching of an element in array
  5. frequency of an element in an array
  6. maximum (largest) element in an array
  7. minimum (smallest) element in an array
  8. Second largest
  9. program to print square of array elements

### Linear search -

Size 6, 11

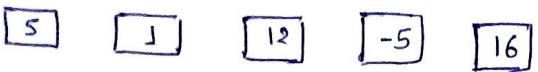
Binary search - Array should be sorted .

variables	first
	last
	mid
	key

10. program to find k<sup>th</sup> element in an array
11. Array element average .
12. • ~~print~~ check array element is even or odd .
13. • print all even and all odd element
13. insert delete .
14. copy array
15. Concept of (mini for loop)



## Bubble sort -



Unsorted

round 1<sup>st</sup>  $i = 0$

j	0	1	2	3	4
	5	1	12	-5	16
1	5	12	-5	16	
1	5	12	-5	16	
1	5	12	-5	16	

$5 > 1$ , swap  $a[0] > a[1]$

5 is not  $> 12$  ok  $a[1]$  not greater  $a[2]$

$12 > -5$   $a[2] > a[3]$  swap

$12 < 16$   $a[3] > a[4]$  swap

round 2<sup>nd</sup>  $i = 1$

j	0	1	2	3	4
	1	5	12	-5	16
1	5	-5	12	16	
1	-5	5	12	16	

$1 < 5$  ok  $a[0] \leq a[1]$

$5 > -5$  swap  $a[1] > a[2]$

$5 < 12$  ok  $a[2] < a[3]$

round 3<sup>rd</sup>  $i = 2$

j	0	1	2	3	4
	1	-5			
		1	5		

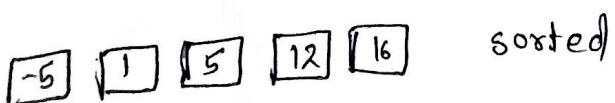
$1 > -5$  swap

$1 < 5$  ok

round 4<sup>th</sup>  $i = 3$

j	0	1	2	3	4
	-5	1			

$-5 < 1$  ok



sorted

```

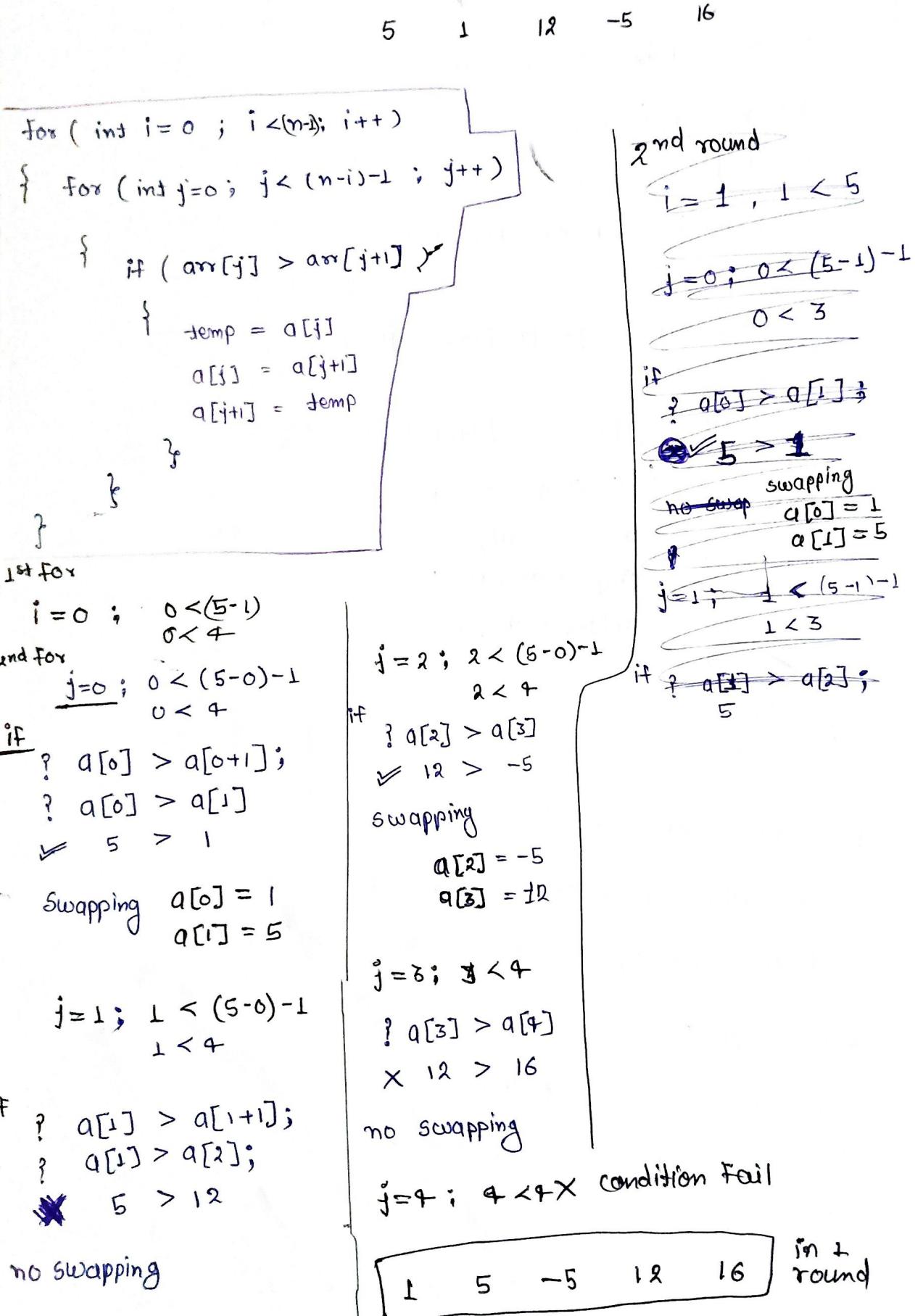
class BubbleSortExample
{
    static void bubbleSort (int [] arr)
    {
        int n = arr.length;
        int temp = 0;

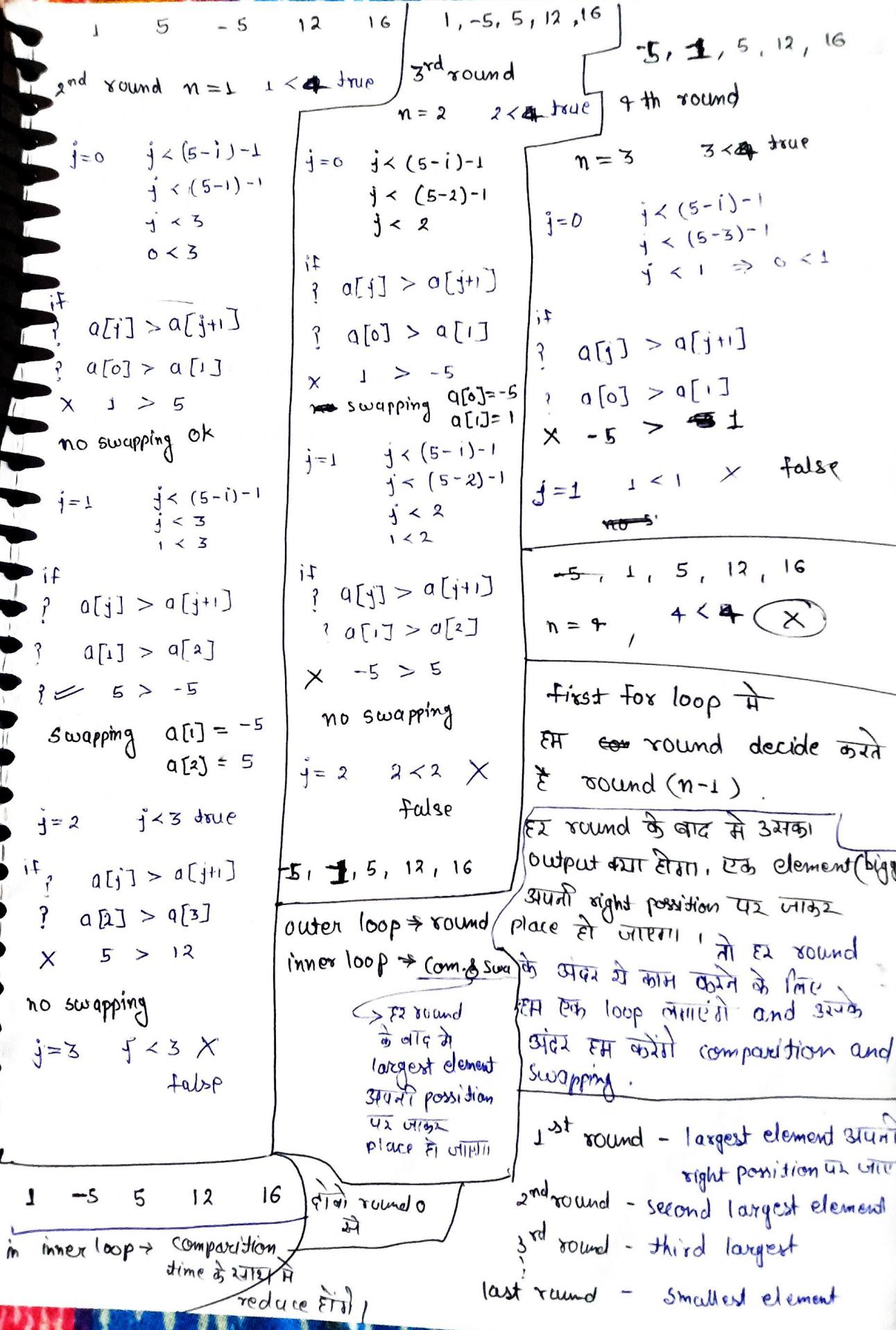
        for ( int i=0 ; i<n-1; i++ )
        {

            for ( int j=0; j<(n-i)-1 ; j++ )
            {
                if ( arr[j] > arr[j+1] )
                {
                    // swap elements
                    temp = arr[j];
                    arr[j] = arr[j+1];
                    arr[j+1] = temp;
                }
            }
        }

        public static void main (String [] args)
        {
            int arr[] = { 3, 60, 35, 2, 45 } ;
            bubbleSort (arr); // sorting array elements, called the method
            System.out.println ("Array after bubble sort");
            for ( int i=0; i<arr.length; i++ )
            {
                System.out.println ( arr[i] + " " );
            }
        }
    }
}

```





## Insertion sort

We use two loops - outer loop  $\rightarrow$  for deciding round  $\Rightarrow (n-1)$   
inner loop  $\rightarrow$  for comparison and swapping

inner loop

$\hookrightarrow$  in bubble sort  $\rightarrow$  inside the inner loop we swap the elements

in Insertion sort  $\rightarrow$  inside the inner loop we shift the elements

in Selection sort  $\rightarrow$  inside the inner loop we swap the elements

$$n = \alpha \cdot \log_{10} 9$$

always

34

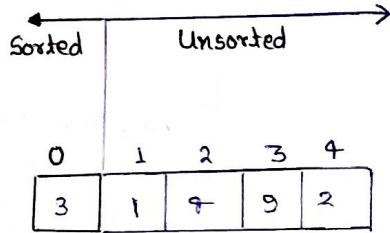
05/05/2020

20

Insertion sort -

if you want to sort any array  
 then firstly consider that the array is already sorted.  
 we make a sorted list  
 we will consider the first element as sorted.  
 Single element is present in list,  
 then this list is definitely sorted.  
 So now we have our sorted list.

and the second list is  
unsorted list



we will take first  
 element from unsorted  
 list.

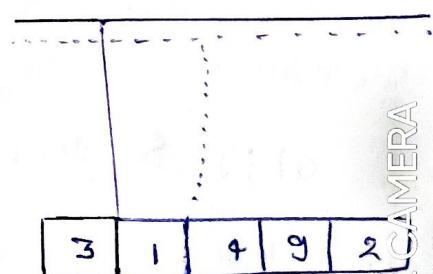
and try to  
 insert it into our  
 sorted list.

now if we are inserting  
 in sorted list.  
 So we will make sure  
 that our element insert in sorted form.  
 because we have to maintain our sorted  
 list sorted itself.

for this purpose, we will not directly  
 insert this element.

we will put this element into a  
 temp. variable.

Until this variable  
 find its exact location.



1
temp

अब  $\downarrow$  को हमने एक temp variable में स्वृति लिया है।

element  $\downarrow$  के लिए जो यही भगाए हैं पहले ये वो उम्मीद index  $\downarrow$  है।

पर इस element  $\downarrow$  को sorted list में ले जाना चाहते हैं,

अब इसके लिए हम उसे sorted list के elements से compare करताएंगे।

अब sorted list का minimum index number 0.

मतलब जब index 0 को प्रोटो होगा तब नहीं करेंगे हम से काम।

so  $j \geq 0$

? अब क्या ये given element आपने आप को हर element compare करेंगा।

नहीं वशोक्ति हम लिखकर compare करवा रहे हैं यह list sorted है।

और sorted ascending order में है।

तो sorted list का last element आगर देखें temp से ~~greater~~ smaller है।

मतलब हमारा temp सबसे बड़ा है,

तो बाकी elements से compare करवाने की जरूरत नहीं है।

$a[j] > \underline{\text{temp}}$

last element  
of sorted  
array

$\text{temp} = 1$

0	1	2	3	4
3	1	4	9	2

outer loop - Unsorted list  
inner loop - sorted list

जब तब  $-1$  ना आ जाए पर हमें साथ एक और condition  $a[j] > \text{temp}$  है। मतलब जब तक हमारा temp द्योता है।

while( $j \geq 0 \ \& \ a[j] > \text{temp}$ )

```

for ( i=1; i<size ; i++ )
{
    temp = a[i];
    j = i-1;

    while ( j >= 0 && a[j] > temp )
    {
        a[j+1] = a[j];
        j--;
    }
    a[j+1] = temp;
}

```

पहले for-loop के जाएंगे

$$i++ \Rightarrow i=2 \quad 2 < 5 \quad \checkmark$$

$$\begin{aligned} \text{temp} &= a[2]; \\ j &= i-1 \Rightarrow 2-1 = 1 \end{aligned}$$

while ( 1 >= 0 && a[j] > temp )

$$\hookrightarrow a[1] > 4$$

$$3 > 4$$

X Condition false

while loop end हो गया

$$a[j+1] = \text{temp};$$

$$a[1+1] = \text{temp}$$

$$a[2] = \text{temp} = 4$$

(जहाँ था वही 28)

अब पापम्

for loop

1	3	4	9	2
---	---	---	---	---

sorted      sorted

$$i++ \Rightarrow 3$$

3	1	4	9	2
---	---	---	---	---

$$i=2 \quad i < 5$$

$$\hookrightarrow \text{temp} = a[1] = 1$$

$$j = i-1 = 1-1 = 0$$

while ( 0 >= 0 && 3 > 1 )

$$\hookrightarrow a[j+1] = a[j];$$

$$a[0+1] = a[0];$$

$$a[1] = a[0];$$

मानक a[1] में a[0] = 3 place

करना है a[1] = 3

अब a[0] में भी अपनी 3 ही है।

3	3	4	9	2
---	---	---	---	---

$$j--; \Rightarrow 0-- \Rightarrow -1$$

$$j >= 0 \Rightarrow -1$$

X false  
while loop से बाहर निकले

Condition false हुयी तो →

line a[j+1] = temp;

$$a[-1+1] = \text{temp}$$

$$a[0] = 1$$

temp मान गया

1	3	4	9	2
---	---	---	---	---

sorted      unsorted

```
for( ; i < size; )
```

$$\text{temp} = a[i] = a[3] = 9$$

$$j = i-1 = 3-1 = 2$$

? while ( $i \geq 0 \ \&\ a[j] > \text{temp}$ )  
 $a[2] > \text{temp}$

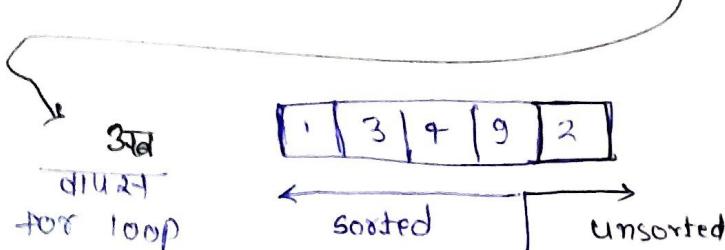
~~4 > 9~~  
condition fail  
while loop के बाहर

$$a[j+1] = \text{temp};$$

$$a[2+1] = \text{temp};$$

$$a[3] = 9$$

जहा तक तकी



$$i++ \Rightarrow 3++ \Rightarrow 4$$

```
for( ; i < size; )
```

$$\text{temp} = a[4] = 2$$

$$j = 4-1 = 3$$

? while ( $i \geq 0 \ \&\ a[j] > \text{temp}$ )  
 $a[3] > \text{temp}$

$$3 > 2$$

true

$$a[3+1] = a[j];$$

$$a[4] = a[3]$$

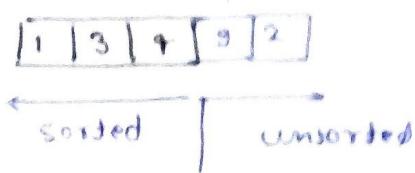
$$a[4] = 3$$

$$j--; \ 3-- \rightarrow 2$$



jump to 9

↑



```
for( i=1; i < size; i++ )
```

$$\{ \text{temp} = a[i];$$

$$j = i-1;$$

while ( $j \geq 0 \ \&\ a[j] > \text{temp}$ )

{

$$a[j+1] = a[j];$$

$$j--;$$

$$a[j+1] = \text{temp};$$

? while ( $i \geq 0 \ \&\ a[j] > \text{temp}$ )  
 $a[2] > \text{temp}$

$$2 > 2$$

true

$$a[2+1] = a[2];$$

$$a[3] = 2;$$

$$j-- \rightarrow 2-- \rightarrow 1$$



→ Post while return

? while ( $i \geq 0$  &  $a[i] > \text{temp}$ )  
 $a[i] > \text{temp}$   
 $3 > 2$   
 $\Leftarrow \text{true}$

$a[j+1] = a[j];$

$a[i+1] = a[i];$

$a[2] = a[1];$

$a[2] = 3$

$j-- \rightarrow i-- \rightarrow 0$

1	2	3	4	9
---	---	---	---	---

for while में जांचता  $j=0$

? while ( $0 \geq 0$  &  $a[0] > \text{temp}$ )  
 $a[0] > 2$   
 $1 > 2$   
 $\times \text{ false}$

$a[j+1] = \text{temp};$

$a[0+1] = \text{temp};$

$a[1] = 2;$

नहीं थी कही छोड़ती ।

अब  
वर्गमूल  
for loop

1	2	3	4	9
---	---	---	---	---

sorted

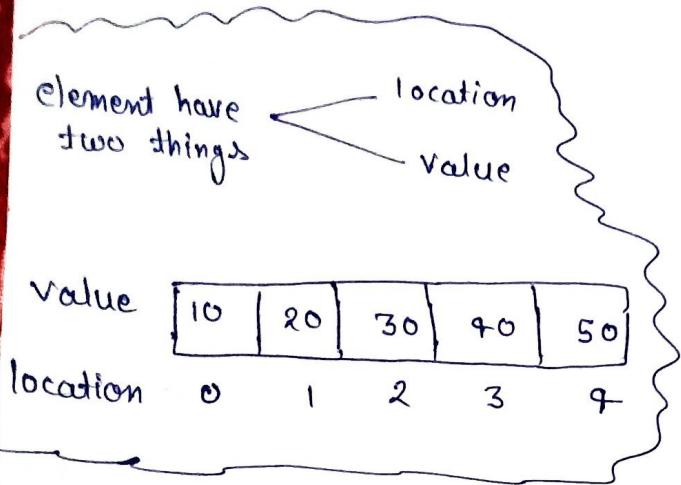
$i++ \Rightarrow 5$

for ( ;  $i < \text{size} ; i++$ )  
 $b < 5 \times$

## Selection sort :-

When we sort our array with the help of selection sort

We first select location 0, and we assume this location's element or declare this location's element as minimum element.



location = index

Outer loop  $\Rightarrow$  round decide

Inner loop  $\Rightarrow$  comparison + swapping

```
for (int i=0 ; i < (size-1); i++)
```

```
{  
    min = arr[i];  
    loc = i;
```

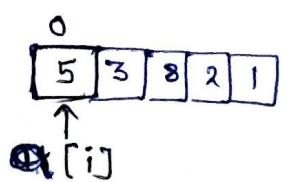
```
    for (j=i+1; j < size; j++)
```

```
{  
    if (arr[j] < min)  
    {  
        min = arr[j];  
        loc = j;  
    }
```

```
    temp = arr[i];
```

```
    arr[i] = arr[loc];
```

```
    arr[loc] = temp;
```



1<sup>st</sup> round →  
 for loop  $i = 0 ; 0 < (5) ; i++$   
 $\min = a[i] = a[0] = 5$   
 $loc = i = 0$

5	3	8	2	1
---	---	---	---	---

2023/03/06 20:35

inner loop 1<sup>st</sup>  
 $j = i+1 = 0+1 = 1 ; j < 5 ; j++$   
 ? if ( $a[j] < \min$ )  
 $a[1] < \min$   
 $3 < \min$   
 $\Leftarrow 3 < 5$   
 $\min = a[j]$   
 $\min = 3$   
 $loc = j = 1$

flow  $j++ \Rightarrow 2 \Rightarrow j < 5$   
 $2 < 5$

? if ( $a[j] < \min$ )  
 $a[2] < \min$   
 $8 < 3$   
 $\times$

flow  $j++ \Rightarrow 2++ \Rightarrow 3 ? 3 < 5$

? if ( $a[3] < \min$ )  
 $\Leftarrow 2 < 3$   
 $\min = a[j]$   
 $\min = 2$   
 $loc = 3 = j$

flow  $j++ \Rightarrow 3++ \Rightarrow 4 ? 4 < 5$

? if ( $a[4] < \min$ )  
 $\Leftarrow 1 < 2$   
 $\min = a[j] = 1$   
 $loc = j = 4$

flow  $j++ \Rightarrow 4++$   
 $j = 5$   
 $? 5 < 5 \times$   
 inner loop end

$temp = a[i]$   
 $temp = 5$   
 $a[i] = a[loc]$   
 $a[0] = \min$   
 $a[0] = 1$   
 $a[loc] = temp$   
 $a[loc] = 5$

after completing this round

1	3	8	2	5
---	---	---	---	---

2<sup>nd</sup> round  $\Rightarrow$

1	3	8	2	5
0	1	2	3	4

$$i = 1 ; i < (5-1); i++$$

$$\min = a[i] = a[1] = 3$$

$$loc = i = 1$$

inner loop ~~not~~

$$j = i+1 ; 2 < 5; j++$$

? if ( $a[j] < \min$ )

$$a[2] < 3$$

$$8 < 3$$

X

$$j++ \Rightarrow$$

$$j = 3 , 3 < 5$$

? if ( $a[j] < \min$ )

$$a[3] < 3$$

$$2 < 3$$

$$\underline{\min = a[j] = 2}$$

$$loc = j = 3$$

$$j++ \Rightarrow j = 4 , 4 < 5$$

? if ( $a[j] < \min$ )

$$a[4] < \min$$

$$5 < 2$$

X

$$j++ \Rightarrow j = 5 , 5 < 5$$

X inner loop ends

$$\text{temp} = a[i];$$

$$a[i] = a[loc]; a[3]$$

$$a[3] = 2$$

$$a[loc] = \text{temp}$$

$$a[3] = 3$$

1	2	8	3	5
0	1	2	3	4

3<sup>rd</sup> round  $\Rightarrow$

$$i = 2 ; 2 < 5$$

$$\min = a[i] = a[3] = 8$$

$$loc = i = 2$$

$$\text{inner loop} - j = i+1 = 2+1 ; i < 5$$

? if ( $a[j] < \min$ )

$$a[4] < \min$$

$$5 < 8$$

$$\min = a[j] = a[4] = 3$$

$$loc = j = 4$$

$$j++ \Rightarrow j = 5$$

1	2	8	3	5
0	1	2	3	4

Round 3rd  $\rightarrow i = 2 ; 2 < (5-1)$

$$\text{inner loop min} = a[i] = a[2] = 8$$

$$\text{loc} = i = 2$$

inner loop  $\text{for } (j=i+1=3$

? if ( $a[j] < \text{min}$ )

$$a[3] < 8$$

$$\swarrow 3 < 8$$

$$\text{min} = a[j] = 3$$

$$\text{loc} = j = 3$$

$$j++ \Rightarrow j=4 \Rightarrow 4 < 5$$

? if ( $a[j] < \text{min}$ )

$$a[4] < \text{min}$$

$$5 < 3$$

~~inner loop~~  $j++ \Rightarrow$   
 $4 \rightarrow 5$

$$\times 5 < 5$$

inner loop ends here

$$\text{temp} = a[i]$$

$$\text{temp} = a[2] = 8$$

$$a[i] = a[\text{loc}]$$

$$a[i] = a[3] = 3$$

$$a[\text{loc}] = \text{temp}$$

$$a[3] = 8$$

1	2	3	8	5
0	1	2	3	4

1	2	3	8	5
0	1	2	3	4

Round 4th  $i = 3 ; 3 < (5-1)$

$$a[i] = a[3] = \text{min} = 8$$

$$\text{loc} = i = 3$$

inner loop  $j = i+1 \Rightarrow j = 4$

? if ( $a[j] < \text{min}$ )

$$5 < 8$$

$$\swarrow \text{min} = a[j] = a[4] = 5$$

$$\text{loc} = j = 4$$

$$j++ \Rightarrow j=5 \quad \cancel{j=5} \quad 5 < 5$$

~~X inner loop ends~~

$$\text{temp} = a[i]$$

$$\text{temp} = 8$$

$$a[i] = a[\text{loc}] = a[4]$$

$$a[5] = 5$$

$$a[\text{loc}] = \text{temp}$$

$$a[5] = 8$$

1	2	3	5	8
0	1	2	3	4

$i++ \Rightarrow i = 4 \quad 4 < (5-1)$

~~X~~

5th round step 2