

Arrays :-

Java: `int [] arr = new int [10];`

C++: `int * arr = new int [10];`

Dono backend pe same hi baat hai.

Ab stack pe banta hai arr naam ka
variable or heap me banta hai array.

Or ab arr me array ke address
store ho jaayega.

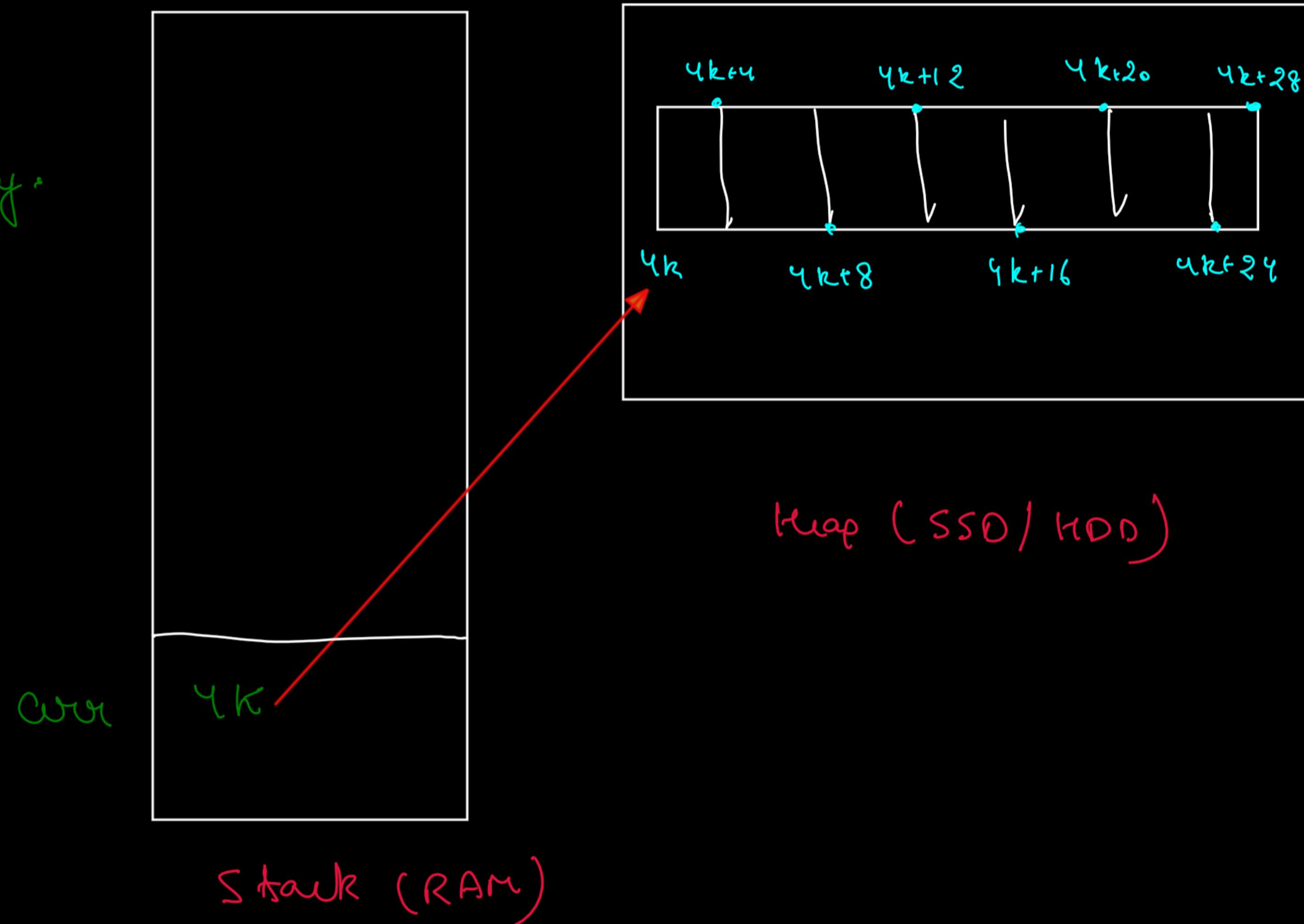
$$arr[2] = 4k + 2(4) \Rightarrow 4k + 8$$

(3rd element)

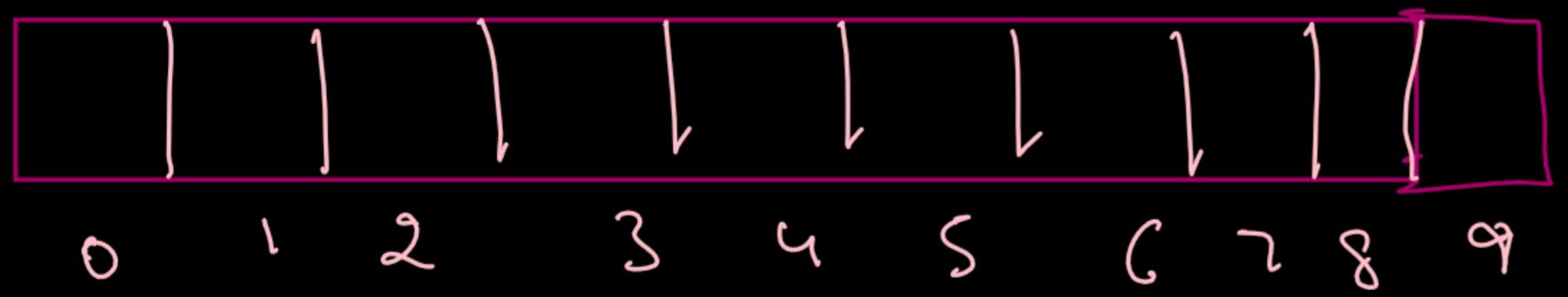
$$arr[4] = 4k + 4(4) \Rightarrow 4k + 32$$

$$arr[0] = 4k + 0(4) \Rightarrow 4k.$$

Jaise hi arr [10] ko use karenge error aa jaayega index out of bounds.



Array indexing



Java me zero base indexing hoti hai.

Agar 6th element ko access karna hoga to likhenge arr[5].

Array ke koi bhi index ki value ko read / access karne me utna hi time lagega jita array ke 0th index me lagta.

Agar stack pe koi variable defined hai or uska data type bhi defined hai to wo primitive data type hote hai. Ex: int, float.

Array non-primitive data type hai

Span of Array: Array ki maximum value - minimum value ko span bolte hai.

```
public static void main(String[] args) throws Exception {
    // write your code here
    Scanner scn = new Scanner(System.in);

    int n = scn.nextInt();
    int[] arr = new int[n];

    for(int i=0;i<arr.length;i++){
        arr[i]=scn.nextInt();
    }

    int max = Integer.MIN_VALUE, min=Integer.MAX_VALUE;

    for(int i=0;i<arr.length;i++){
        max = Math.max(max,arr[i]);
        min = Math.min(min,arr[i]);
    }

    int span = max-min;
    System.out.println(span);
}
```

$$\text{Max} = \text{Integer.MIN_VALUE}$$

can be written as

$$\text{Max} = (\text{int}) - 1e8$$

$$1e8 = 10^8$$

Similarly,

$$\text{Min} = (\text{int}) 1e8 -$$

Linear Search: Puri array pe traverse karke value dhundte hai

```
public static int linearSearch(int [] arr, int target){  
    for(int i=0;i<arr.length;i++){  
        if(arr[i]==target){  
            return i;  
        }  
    }  
    return -1;  
}  
  
public static void main(String[] args) throws Exception {  
    Scanner scn = new Scanner(System.in);  
  
    int n = scn.nextInt();  
    int[] arr = new int[n];  
  
    for(int i=0;i<arr.length;i++){  
        arr[i]=scn.nextInt();  
    }  
  
    int target=scn.nextInt();  
  
    int index=linearSearch(arr,target);  
    System.out.println(index);  
}
```

Bar Chart: Array ki maximum value ko use karke har ek value se compare karke check karna ki value \geq maximum agar true hota hai to '*' print nahi to '\t'.

Example

Sample Input

```
5  
3  
1  
0  
7  
5
```

Sample Output

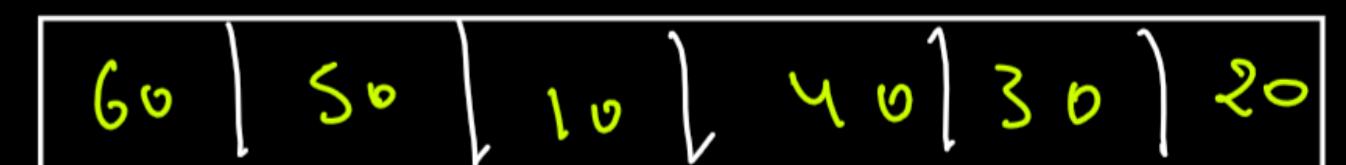
```
*  
*  
*  *  
*  *  
*  *  
*  *  
*  *
```

```
for (int i = 0; i < n; i++) {  
    arr[i] = scn.nextInt();  
}  
  
int max = Integer.MIN_VALUE;  
  
for (int i = 0; i < n; i++) {  
    if (arr[i] > max) {  
        max = arr[i];  
    }  
}  
  
for (int i = max; i > 0; i--) {  
    for (int j = 0; j < n; j++) {  
        if (arr[j] >= i) {  
            System.out.print("*\t");  
        } else {  
            System.out.print("\t");  
        }  
    }  
    System.out.println();  
}
```

Reverse of Array :- Ek pointer ko 0th index pe or ek ko last pe rakh kar swap kardenge then left ++, right --; tab tak karenge jab tak left < right.

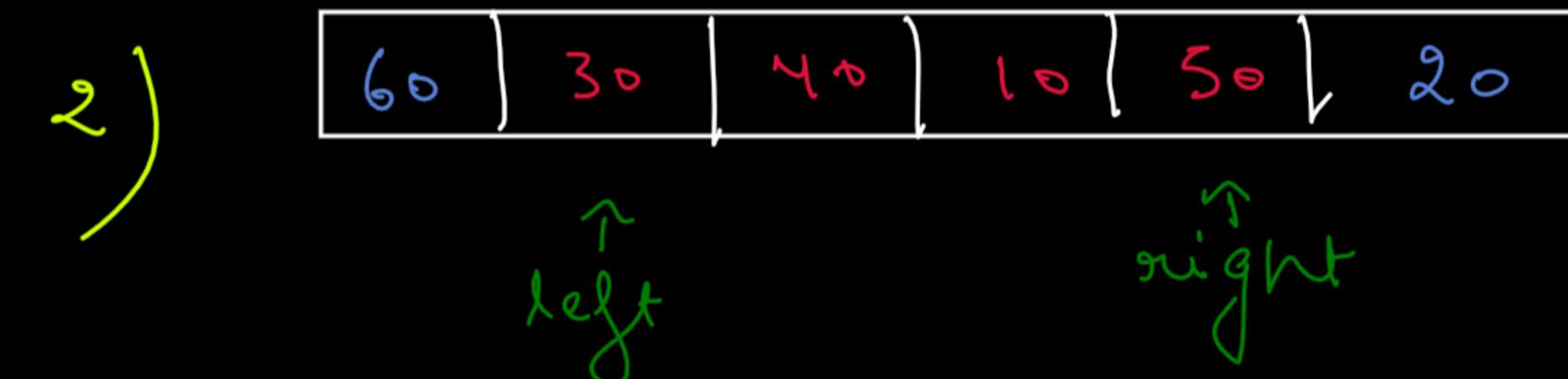
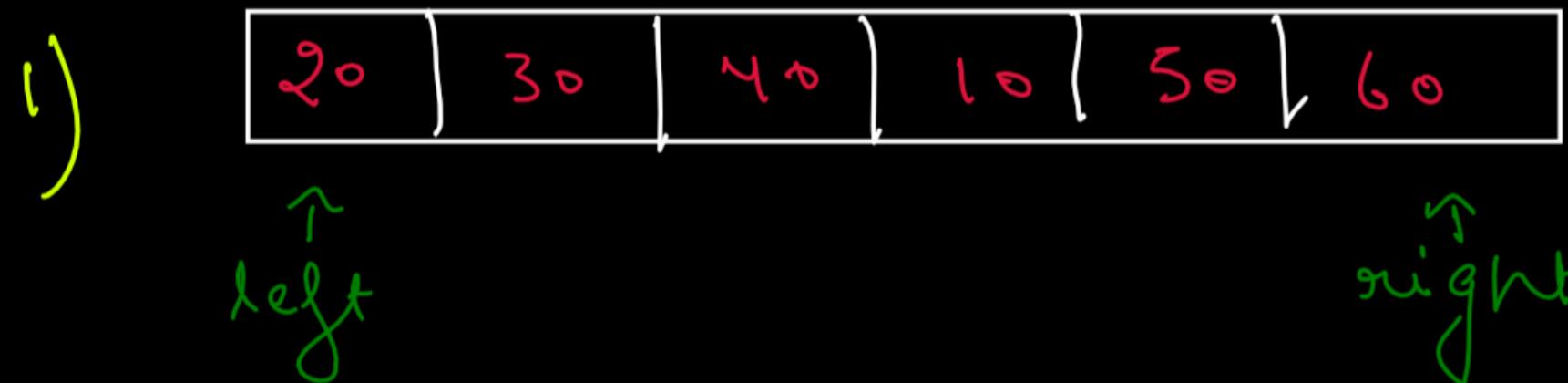
```
public static void swap(int [] arr,int left,int right){
    int temp = arr[left];
    arr[left]=arr[right];
    arr[right]=temp;
}

public static void reverse(int[] a){
    // write your code here
    int left=0;
    int right=a.length-1;
    while(left<right){
        swap(a,left,right);
        left++;
        right--;
    }
}
```



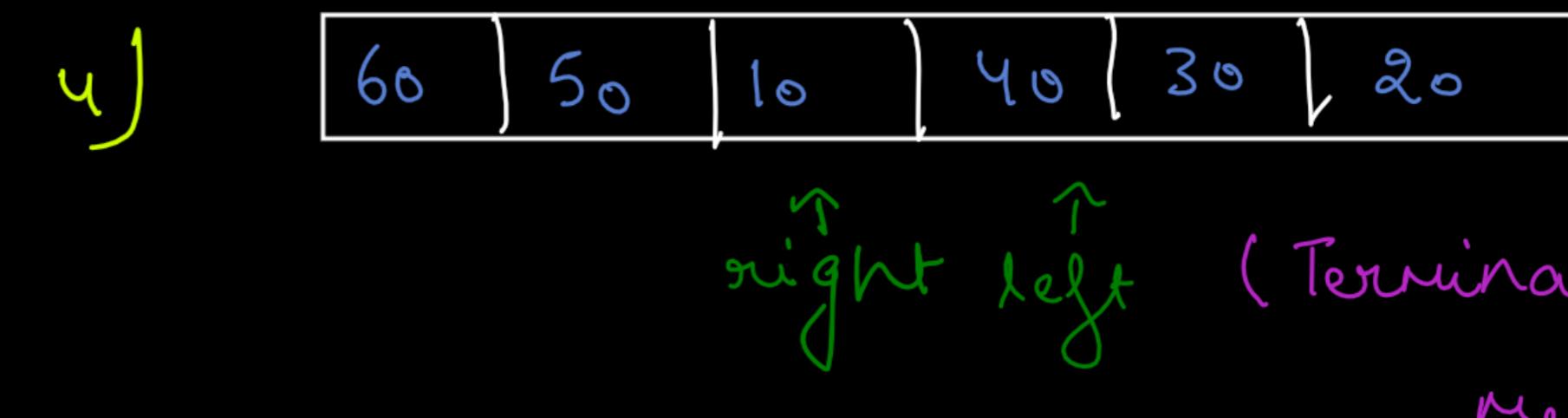
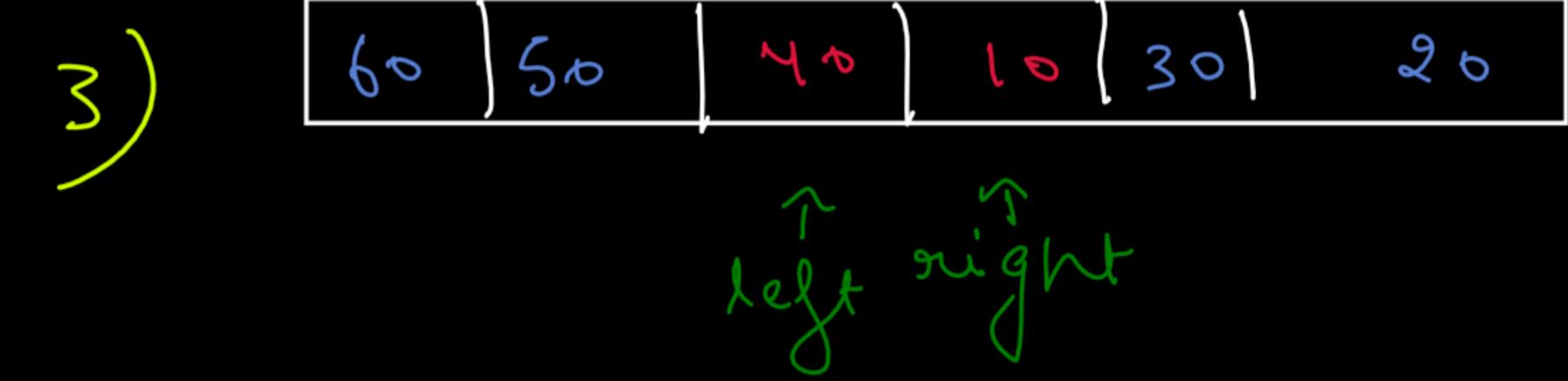
Reverse

Explanation:



while (left < right)

```
{
    swap( arr, left, right );
    left++;
    right--;
}
```



Sum of 2 Arrays: 2 arrays given hoga dono ke last se iterate karte hue 0th index tak ki values ka sum nikalna hai or ek new array me push karna hai

$$\begin{array}{r} 999 \\ + 01 \\ \hline 1000 \end{array}$$

$$\begin{array}{r} 1234 \\ + 1111 \\ \hline 2345 \end{array}$$

$$\begin{array}{r} ① \\ 4763 \\ + 1\cancel{3}34 \\ \hline 6097 \end{array}$$

Jo answer array hai uska size humesha jo 2 given arrays hai usme se jo bada hai uske equal hoga. Because atleast jitni terms badi array re hoga utni terms to answer array me hoga hi. Atmost ek term extra aa sakti hai or vo term bhi carry handle kar lega. For ex:-

$$\begin{array}{r} ① ① ① \\ 999 \\ + 01 \\ \hline 1000 \end{array}$$

↑ ↘ 1

left over
carry → answer array

Code :-

```

int[] res = new int[n1 > n2 ? n1 : n2];

int k = res.length - 1;
int j = arr2.length - 1;
int i = arr1.length - 1;

int carry = 0;

while (k >= 0) {
    int sum = 0;
    sum += carry;

    if (i >= 0) {
        sum += arr1[i];
    }
    if (j >= 0) {
        sum += arr2[j];
    }

    res[k] = sum % 10;
    carry = sum / 10;
    i--;
    k--;
    j--;
}

if(carry>0){
    System.out.println(carry);
}

for(int val : res){
    System.out.println(val);
}

```

$$\rightarrow k = -1, i = -1, j = -3$$

Loop terminate hua and carry ne l bacha hai
abhi

or ans array hai [0 0 0]

Dry Run :-

arr1 :-

$$\begin{array}{|c|c|c|} \hline 9 & 9 & 9 \\ \hline 0 & 1 & 2 \\ \hline
 \end{array}
 + \begin{array}{|c|} \hline 1 \\ \hline 0 \\ \hline
 \end{array}
 \underline{\quad}$$

arr2 :-

$$\begin{array}{|c|c|c|} \hline 1 & 0 & 0 \\ \hline 0 & 0 & 0 \\ \hline
 \end{array}
 \begin{array}{l} \text{carry} \leftarrow \\ \text{answer} \end{array}
 \begin{array}{l} \uparrow \\ \uparrow \end{array}
 \begin{array}{l} \text{array} \\ \text{array} \end{array}$$

$$\rightarrow k = 2, i = 2, j = 0$$

$$\text{sum} = 10, \text{ans}[k] = 10 \% 10 = 0$$

$$\text{carry} = 10 / 10 = 1$$

$$\rightarrow k = 1, i = 1, j = -1$$

$$\text{sum} = \frac{9+1}{10} \times \text{carry} = 10, \text{ans}[k] = 10 \% 10 = 0$$

$$\text{carry} = 10 / 10 = 1$$

$$\rightarrow k = 0, i = 0, j = -2$$

$$\text{sum} = 10, \text{ans}[k] = 10 \% 10 = 0$$

$$\text{carry} = 10 / 10 = 1$$

Difference of 2 Arrays: Array 2 me se array 1 ko minus karke ek new array me value push karne hai.

$$\begin{array}{r} Y: \quad \begin{array}{r} 9 \\ 806 \\ - 8 \\ \hline 798 \end{array} \\ X: \quad \end{array}$$

$$\begin{array}{r} Y: \quad \begin{array}{r} 841 \\ - 1 \\ \hline 840 \end{array} \\ X: \quad \end{array}$$

$$\begin{array}{r} Y: \quad \begin{array}{r} 1000 \\ - 1 \\ \hline 999 \end{array} \\ X: \quad \end{array}$$

Answer array ka size hamesha array Y (or B) ke equal hi hogा kyuki question me given hai array Y hamesha bada hogा aur X se and subtraction me resultant array kabhi bhi exceed nahi karega array Y ke size se

Code: Main

```
Scanner scn = new Scanner(System.in);
int n1 = scn.nextInt();
int [] A = new int[n1];
for(int i = 0; i<n1; i++){
    A[i] = scn.nextInt();
}
// n2 > n1
int n2 = scn.nextInt();
int [] B = new int[n2];
for(int i = 0; i<n2; i++){
    B[i] = scn.nextInt();
}

int [] ans = difference(A,B);
//edge cases
// 1. first non zero index -- fnzi
int fnzi = -1;
for(int i = 0; i<ans.length; i++){
    if(ans[i] != 0){
        fnzi = i;
        break;
    }
}
//546-546 = 000
if(fnzi == -1){
    System.out.println("0");
    return;
}
for(int i = fnzi; i<ans.length; i++){
    System.out.println(ans[i]);
}
```

difference
function →

```
public static int [] difference(int [] A, int [] B){
    int borrow = 0;
    int [] ans = new int[B.length];
    int i = A.length-1;
    int j = B.length-1;
    int k = ans.length-1;

    while(j>=0){
        int digit = 0;
        digit = B[j] + borrow;
        if(i >= 0){
            digit -= A[i];
        }

        if(digit < 0){
            digit += 10;
            borrow = -1;
        } else{
            borrow = 0;
        }

        int val = digit;
        ans[k] = val;
        i--;
        j--;
        k--;
    }
    return ans;
}
```

Dry Run:

$$\begin{array}{r} 5006 \\ 76 \\ \hline 4930 \end{array}$$

1) $j=3, i=1, borrow=0$

$digit = 6$

$digit -= 6 \Rightarrow digit = 0$

$ans[3] = 0$

2) $j=2, i=0, borrow=0$

$digit = 0 + 0 = 0$

$digit -= 7 = -7$

$digit < 0$

$digit += 10 \Rightarrow digit = 9$

$borrow = -1$

$ans[2] = 3$

3) $j=1, i=-1, borrow=-1$

$digit = 0 - 1 \Rightarrow -1$

$digit < 0$

$digit += 10 \Rightarrow digit = 9$

$borrow = -1$

$ans[1] = 9$

4) $j=0, i=-2, borrow=-1$

$digit = 5 - 1 = 4$

$ans[0] = 4$

Rotate an Array: Array pe 3 baar reverse function lagana hai.

- 1) 0 to length - k - 1.
- 2) length - k to length - 1.
- 3) 0 to length - 1.

Suppose ek array hai: A:

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

ab array A ko agar 3 se
rotate karenge to output
banega

\downarrow
 $k = 3$

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

Last ke 'k' elements

starting me chale jaayenge
and rest cage shift ho
jaayenge

$k=0, 5 \rightarrow$ 1 2 3 4 5

Jab bhi $k \geq$ arr. length

$k=1, 6 \rightarrow$ 5 1 2 3 4

to cases repeat hone

$k=2, 7 \rightarrow$ 4 5 1 2 3

lagerge.

$k=3, 8 \rightarrow$ 3 4 5 1 2

for ex: $k=0$ is same as

$k=4, 9 \rightarrow$ 2 3 4 5 1

$k=5$.

$k=5 \rightarrow$ 1 2 3 4 5

So, K can be written $k = k \% \text{A.length}$

$k=-1, 4 \rightarrow$ 2 3 4 5 1

and K jab -ive hai to 1) $k = k \% \text{A.length}$.

Same as
 $k=4$

$k=-2, 3 \rightarrow$ 3 4 5 1 2

2) $k = k + \text{A.length}$.

Code :

```
public static void reverse(int[] arr, int left,int right){  
    while(left<right){  
        int temp = arr[left];  
        arr[left] = arr[right];  
        arr[right] = temp;  
  
        left++;  
        right--;  
    }  
}  
public static void rotate(int[] a, int k){  
    k= k % a.length;  
  
    if(k<0){  
        k = k + a.length;  
    }  
  
    int n = a.length;  
    reverse(a, 0, n-k-1);  
    reverse(a, n-k, n-1);  
    reverse(a, 0, n-1);  
}
```

Dry Run :-

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

1) 0th index se length - l - k me reverse

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

2) length - k se length - l me reverse

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

3) 0 se length - l me reverse

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

Inverse of Array :- Value or index ko interchange karna hai.

Array A :-

4	0	2	3	1
0	1	2	3	4

Code :-

```
public static int[] inverse(int[] a){
    int [] ans = new int [a.length];

    for(int i=0; i<a.length; i++){
        int val = a[i];
        ans[val] = i;
    }

    return ans;
}
```

Inverse of A :-

1	4	2	3	0
0	1	2	3	4

Dry Run :-

Inverse Of An Array

A = $\begin{bmatrix} 1 & 2 & 0 & 4 & 3 \end{bmatrix}$ \longleftrightarrow ans $\begin{bmatrix} 2 & 0 & 1 & 4 & 3 \end{bmatrix}$

Some new array banake wo return hogi because agar same array me iterate karenge to LOSS OF INFORMATION hogta.

int [] ans = new int [A.length];
for (int i=0; i<A.length; i++){
 int value = A[i];
 ans[value] = i;
}

Dry RUN :- $\rightarrow i=0$, value = 1
ans[1] = 0.

$\rightarrow i=1$, value = 2
ans[2] = 1.

$\rightarrow i=2$, value = 0
ans[0] = 2.

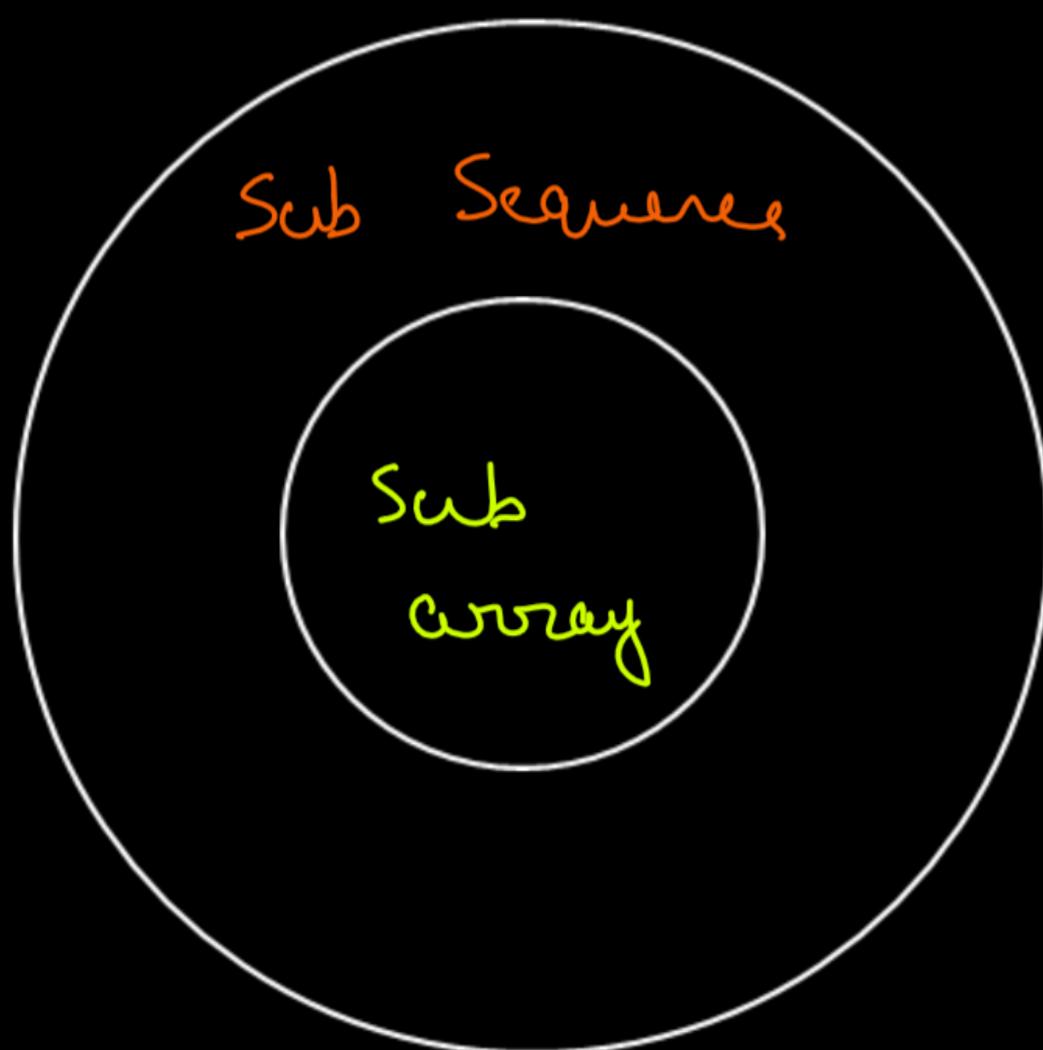
$\rightarrow i=3$, value = 4
ans[4] = 3.

$\rightarrow i=4$, value = 3
ans[3] = 4.

0				
0	1			
2	0	1		
2	0	1	3	
2	0	1	4	3

- Subarray: 1) Subarray is continuous segment of array.
 on
Substring 2) Subarray is unidirectional.
 3) For an array of size n , the total no. of subarrays are $\frac{n(n+1)}{2}$

- Subset or Subsequence: 1) It may or may not continuous.
 (Binary Relation) 2) It is unidirectional
 3) If size of array is n then total no. of subsequence : 2^n .



Subarray Question :

```

public static void subArray(int[] arr) {
    for (int i = 0; i < arr.length; i++) {
        for (int j = i; j < arr.length; j++) {
            for (int k = i; k <= j; k++) {
                System.out.print(arr[k] + "\t");
            }
            System.out.println();
        }
    }
}

```

SubArray

Input : 10 20 30	Output : 10
	10 20
	10 20 30
	20
	20 30
	30

Pseudo Code :-

array A :

10	120	30
----	-----	----

```

for (int i=0; i < A.length; i++) {
    for (int j=i; j < A.length; j++) {
        for (int k=i; k <= j; k++) {
            System.out.print(A[k] + " space");
        }
    }
}

```

Dry Run :-

$\rightarrow i=0, j=0,$

$\rightarrow i=0, j=1, \quad k=0, \text{ print } (10 + \text{" space"}) ;$

$\quad \quad \quad k=0, \text{ print } (10 + \text{" space"}) ;$

$\quad \quad \quad k=1, \text{ print } (20 + \text{" space"}) ;$

$\rightarrow i=0, j=2,$

$\quad \quad \quad k=0, \text{ print } (10 + \text{" space"}) ;$

$\quad \quad \quad k=1, \text{ print } (20 + \text{" space"}) ;$

$\quad \quad \quad k=2, \text{ print } (30 + \text{" space"}) ;$

$\rightarrow i=1, j=1$

$\quad \quad \quad k=1, \text{ print } (20 + \text{" space"}) ;$

$\rightarrow i=1, j=2$

$\quad \quad \quad k=1, \text{ print } (20 + \text{" space"}) ;$

$\quad \quad \quad k=2, \text{ print } (30 + \text{" space"}) ;$

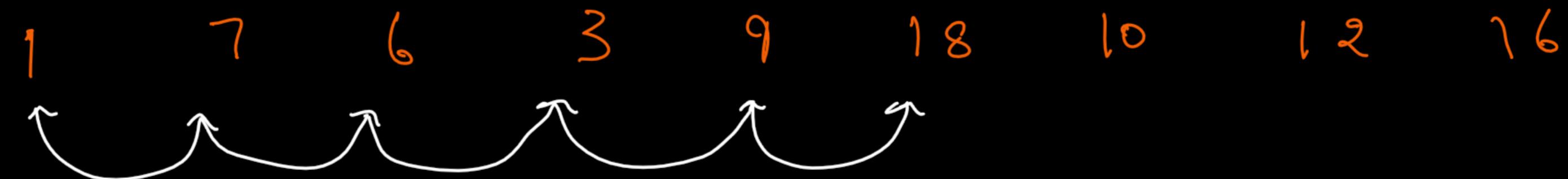
$\rightarrow i=2, j=2$

$\quad \quad \quad k=2, \text{ print } (30 + \text{" space"}) ;$

Binary Search :

Linear

Search :

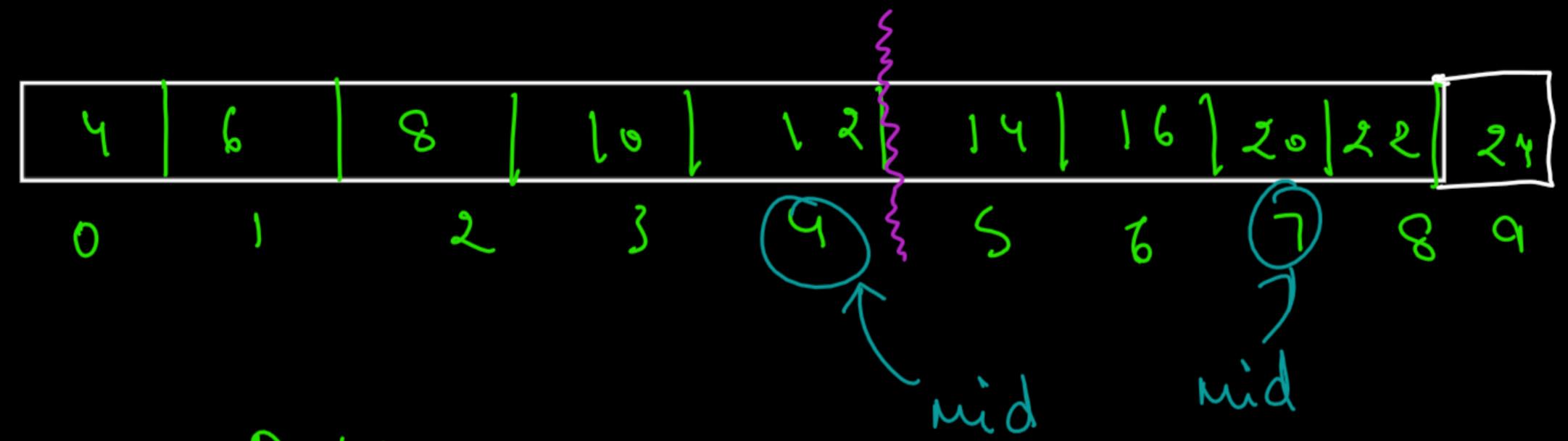


for $x=18$.

- Linear Search me Puri array pe traverse karenge or ek ek karke values check karte raha hoga.
- Agar 9 elements huu array me or last element humne find karna hua to 9 operations lagenge (traversals) Similarly 10^6 huu elements to 10^6 baar traverse karna hoga.
- Searching ko optimise karne ke liye binary search algo use karunge.

Binary Search: 1) Array should be sorted.

To find: $x = 20$



① $\text{mid} = \frac{0+9}{2} = 4$

→ Kya 20 arr[mid] se bada hai?

20 bada hai arr[mid] se to palka
vo mid ke baad vale region nu
lie karega.

② $\text{mid} = \frac{5+9}{2} = 7 \Rightarrow \text{arr[mid]} == 20.$

Dry Run:

→	<table border="1"> <tr><td>4</td><td>6</td><td>8</td><td>10</td><td>12</td><td>14</td><td>16</td><td>18</td><td>20</td><td>22</td><td>24</td></tr> <tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td></tr> </table>	4	6	8	10	12	14	16	18	20	22	24	0	1	2	3	4	5	6	7	8	9	10	for $x = 4$.
4	6	8	10	12	14	16	18	20	22	24														
0	1	2	3	4	5	6	7	8	9	10														

$$\rightarrow \text{mid} = (0+10)/2 = 5.$$

is $\text{arr}[\text{mid}] == x$ X

is $k > \text{arr}[\text{mid}]$ X

else $k < \text{arr}[\text{mid}]$

$$\text{right} = \text{mid} - 1 = 4$$

$$\leftarrow \text{mid} = (0+4)/2 = 2$$

is $\text{arr}[\text{mid}] == x$ X

is $k > \text{arr}[\text{mid}]$ X

else $k < \text{arr}[\text{mid}]$

$$\text{right} = \text{mid} - 1 = 1.$$

$$\rightarrow \text{mid} = \frac{(0+1)}{2} = 0$$

is $\text{arr}[\text{mid}] == x$

return mid;

```
public static int binarysearch(int [] arr, int k){
    // code here
    int left = 0, right = arr.length-1;

    while(left<=right){
        int mid = (left+right)/2;
        if(arr[mid]==k){
            return mid;
        }
        else if(arr[mid]<k){
            left=mid+1;
        }else{
            right=mid-1;
        }
    }
    return -1;
}
```

4	6	8	10	12	14	16	18	20	22	24
0	1	2	3	4	5	6	7	8	9	10

for $x = 11$.

$$\rightarrow \text{mid} = (0+10)/2 = 5$$

is $\text{arr}[\text{mid}] == x \times$

is $\text{arr}[\text{mid}] < x$

else $x < \text{arr}[\text{mid}]$

$$\text{right} = \text{mid} - 1 = 4$$

$$\rightarrow \text{mid} = (0+4)/2 = 2$$

is $\text{arr}[\text{mid}] == x \times$

is $x > \text{arr}[\text{mid}]$

$$\text{left} = \text{mid} + 1 = 3$$

$$\rightarrow \text{mid} = \frac{(3+4)}{2} = 3$$

is $\text{arr}[\text{mid}] == x \times$

is $x > \text{arr}[\text{mid}]$

$$\text{left} = \text{mid} + 1 = 4$$

$$\rightarrow \text{mid} = (4+4)/2 = 4$$

is $\text{arr}[\text{mid}] == x \times$

is $\text{arr}[\text{mid}] < x$

else $\text{arr}[\text{mid}] > x$

$$\text{right} = \text{mid} - 1 = 3$$

Loop terminate ho jaayega because

$\text{left} > \text{right}$

$\text{return } -1;$

Broken Economy :-

1	5	10	15	22	33	40	42	55	66
0	1	2	3	4	5	6	7	8	9

$x = 34$

Approach

$$\rightarrow \text{mid} = \frac{(0+9)}{2} = 4 \quad \rightarrow \quad \text{mid} = \frac{(5+9)}{2} = 7$$

$\text{arr}[\text{mid}] = ? \quad \times$

$\text{Floor} = 2 \cancel{2} \quad 33$

$\text{ceil} = 4 \cancel{2}, 40$

$22 < 34$

$\text{left} = \text{mid} + 1$

$42 > 34$

$\text{right} = \text{mid} - 1.$

$$\rightarrow \text{mid} = \frac{(5+6)}{2} = 5$$

$33 < 34$

$\text{left} = \text{mid} + 1.$

$$\rightarrow \text{mid} = \frac{(6+6)}{2} = 6$$

$40 > 34$

$\text{right} = \text{mid} - 1.$

Dry Run:

Broken Economy

Ceil and floor vala question hai. Agar given values exist karegi array me to ceil = floor = value else use just badi value jo exist karegi array me wo ceil and chahi value floor me store hoga.

Ex:

10	12	13	25	28
----	----	----	----	----

 Value: 19
ceil = 25, floor = 13.

Dry Run :
 $l_i = 0, 3$ $u_i = 0$
 $r_i = 4, 2$ $f_{loor} = 0, 13$
 $\rightarrow mid = (0+4)/2 = 2$
 is $(arr[mid] == 19)$?
 $arr[mid] < 19$
 $l_i = mid+1 ; f_{loor} = arr[mid].$
 $\rightarrow mid = (3+4)/2 = 3$
 is $(arr[mid] == 19)$?
 $arr[mid] > 19$
 $r_i = mid-1 , \quad ceil = arr[mid].$

$l_i > r_i$ condition falls loop break hoga then
ceil and floor print.

```

public static void ceilFloor(int [] arr , int k){
    int left = 0, right = arr.length-1;
    int floor = 0, ceil = 0;

    while(left<=right){

        int mid = (left+right)/2;

        if(arr[mid]==k){

            floor = arr[mid];
            ceil = arr[mid];
            break;

        }else if(k>arr[mid]){

            floor = arr[mid];
            left = mid+1;

        }else if(k<arr[mid]){

            ceil = arr[mid];
            right = mid-1;

        }

    }

    System.out.println(ceil);
    System.out.println(floor);
}
  
```

First & Last Index:

```
public static int firstIndex(int [] arr, int n){  
    int li=0;  
    int ri=arr.length-1;  
    int index = -1;  
    while(li<=ri){  
        int mid = (li + ri) / 2;  
  
        if(arr[mid]==n){  
            index=mid;  
            ri=mid-1;  
        }else if(n>arr[mid]){  
            li=mid+1;  
        }else{  
            ri=mid-1;  
        }  
    }  
    return index;  
}  
  
public static int lastIndex(int [] arr, int n){  
    int li=0;  
    int ri=arr.length-1;  
    int index = -1;  
    while(li<=ri){  
        int mid = (li + ri) / 2;  
  
        if(arr[mid]==n){  
            index=mid;  
            li=mid+1;  
        }else if(n>arr[mid]){  
            li=mid+1;  
        }else{  
            ri=mid-1;  
        }  
    }  
    return index;  
}
```

Binary Search

```
public static int firstIndex(int [] arr, int n){  
    int ind=-1;  
  
    for(int i=0;i<arr.length;i++){  
        if(arr[i]==n){  
            ind=i;  
            break;  
        }  
    }  
    return ind;  
}  
  
public static int lastIndex(int [] arr, int n){  
    int ind=-1;  
  
    for(int i=arr.length-1 ;i>=0; i--){  
        if(arr[i]==n){  
            ind=i;  
            break;  
        }  
    }  
    return ind;  
}
```

Linear Search

Dry Run:

First and last Index										
0	1	2	3	4	5	6	7	8	9	10

For 30 → First index = 5, last index = 7

→ First Index.

index = -X 5
 $l_i = 0 \neq X 5$
 $r_i = 10 \neq 4$

→ mid = $(0+0)/2 = 5$.

is $(arr[5] == 30)$?
 index = 5, mi = mid-1;

Since first index change to go to left side me hi exist karega to current value ka left me binary search phir lega denge

→ mid = $(0+4)/2 = 2$
 is $(arr[2] == 30)$?
 $arr[2] < 30$
 $l_i = mid+1$.

→ mid = $(3+4)/2 = 3$
 is $(arr[3] == 30)$?
 $arr[3] < 30$
 $l_i = mid+1$.

→ mid = $(4+4)/2 = 4$
 is $(arr[4] == 30)$?
 $arr[4] < 30$.

$l_i = mid+1$,
 $l_i > r_i$ loop break hoga
 and jo bhi index ki value hai vo print ho jayegi.

→ Last Index.

index = -X 5 & 7
 $l_i = 0 \neq 7$
 $r_i = 10 \neq 7$

→ mid = $(0+10)/2 = 5$
 is $(arr[mid] == 30)$?
 index = mid; $l_i = mid+1$;

Since last index find karna hai to current index ke right side me hi exist karega to ab right side me binary search lega denge.

→ mid = $(6+10)/2 = 8$
 is $(arr[mid] == 30)$?
 $arr[mid] > 30$
 $mi = mid-1$.

→ mid = $(6+7)/2 = 6$
 is $(arr[mid] == 30)$?
 index = mid; $l_i = mid+1$;

→ mid = $(7+7)/2 = 7$
 is $(arr[mid] == 30)$?
 index = mid, $l_i = mid+1$;

$l_i > r_i$ condition falla hogayi loop break hoga and index print.

Subset :

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

- - - 0 0 0
 - - 30 0 0 1
 - 20 -
 - 20 30 ↗ 0 1 0
 10 - -
 10 - 30 0 1 1
 10 20 -
 10 20 30 1 0 0
 1 0 1
 1 1 0
 1 1 1

Approach : total subsets = 2^n , $n \rightarrow$ length of array.

```
for (i=0 ; i<2n ; i++) {
    {
        i=0      convert Binary
        if (rem==0)
        {
            print (" - ");
        }
        else
        {
            print (arr[i] + " ");
        }
    }
}
```

```
public static void main(String[] args) {
    Scanner scn= new Scanner(System.in);
    int n = scn.nextInt();
    int [] arr = new int[n];
    for(int i=0;i<n;i++){
        arr[i]=scn.nextInt();
    }
    for(int i=0 ; i<(int)Math.pow(2,n);i++){
        int temp = i;
        String str="";
        int index=arr.length-1;
        while(index>=0){
            int rem = temp%2;
            temp/=2;
            if(rem==1){
                str = arr[index]+\t+str;
            }else{
                str = "-\t"+str;
            }
            index--;
        }
        System.out.println(str);
    }
}
```

Subset

If size of array is 'n' then total no. of subsets = 2^n .

A =

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

 Subsets

- - -
- * - 30
- 20 -
- 20 30
10 - -
10 - 30
10 20 -
10 20 30

Dry Run :- $\rightarrow i=0, i < 8$

temp = 0, strn = "", index = $2^0 \times 2^{0-1}$.

num = 0, temp = 0

strn = "-" + strn

$\rightarrow i=1, i < 8$

* temp = 0, strn = "", index = 2.

num = 1, temp = 0

strn = " 30"

* temp = 0, strn = "30", index = 1.

num = 0, temp = 0

strn = "-" + "30".

* temp = 0, strn = "- 30", index = 0

num = 0, temp = 0

strn = "- " + "- 30".

print (strn);

$\rightarrow i=2, i < 8$

temp = 2, strn = "", index = 2.

num = 0, temp = 1.

strn = " - "

temp = 1, strn = "+ ", index = 1.

num = 1, temp = 0

strn = "20" + "+ - "

temp = 0, strn = "20 - ", index = 0

num = 0, temp = 0

strn = "- " + strn.

print (strn);

$\rightarrow i=3, i < 8$

temp = 3, index = 2, strn = "".

num = 1, temp = 1

strn = " 30";

temp = 1, index = 1, strn = "30".

num = $1 \times 2 = 1$, temp = $1/2 = 0$

strn = "20" + "30".

temp = 0, index = 0, strn = "20 30".

num = 0, temp = 0

strn = "- " + strn.

print (strn);