

Arrays & ArrayList

M	T	W	T	F	S	S
Page No.:	YOUVA					
Date:						

① Linear Search:-

Q. Find the element 'x' in the array. Take array & 'x' as input?

⇒ Code :- Psvm {

```

Scanner sc = new Scanner(System.in);
Sout("Enter target Element : ");
int x = sc.nextInt();           // x = target element.
Sout("Enter Array Size : ");
int n = sc.nextInt();           // n = size of array taken as a input.
int[] arr = new int[n];
Sout("Enter Array Element");
for (int i = 0; i < n; i++) {
    arr[i] = sc.nextInt();
}
    } Taking array as input from user
  }
```

// Solution

boolean flag = false; // false means Not found.

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

if (arr[i] == x) {

flag = true; // true means found.

break;

}

.

if (flag == true) Sout("Element Found");

else Sout("Element not Found");

}

⇒ Boolean flag used for optimization. Whenever if condition satisfied Break the loop.

④ buildIn Method:-

⇒ Code :- Psvm {

int[] arr = {60, 50, 20, 30, 40, 10};

Arrays.sort(arr); } Building B method to sort array

for (int ele : arr) {

Sout(ele + " "); } }

import java.util.Arrays;

To use this we have to import this

} For Each Loop. Use when you don't need indexes.

② Maximum & Second Maximum Element.

M	T	W	T	F	S	S
Page No.:	YOUVA					
Date:						

⇒ Code :- Psvm {

int [] arr = {23, 34, 53, 56, 63, 64, 34, 34, 2, 25, -45}; // Easiest way of creating array
int n = arr.length;

{ int mx = Integer.MIN_VALUE;
for (int i=0; i<n; i++) {
 // if (arr[i] > mx) mx = arr[i]; } without using built-in function
 mx = Math.max(mx, arr[i]); } This is using built-in Math function.
}.

{ int smx = Integer.MIN_VALUE;
for (int i=0; i<n; i++) {
 // if (arr[i] > smx && arr[i] != mx) smx = arr[i];
 if (arr[i] != mx)
 smx = Math.max(smx, arr[i]); }
}.

Sout(smx);

}

⇒ To find second max you have to find maximum first.

⇒ Integer.MIN_VALUE; } Stores the smallest possible ^{int} value.
Integer.MAX_VALUE; } Stores the highest possible ^{int} value.

⇒ Math.max(smx, arr[i]); } An build in function to find max value b/w to element.

Math.min(smx, arr[i]); } An build in function to find min value b/w to element.

③ Passing Arrays to Methods

⇒ Output :- 10

⇒ Code :- Psvm { int[] arr = {10, 20, 30, 40};

• 90

Sout(arr[0]);

| # Because in Arrays it pass by Reference Variable.

Change(arr);

| Reference Variable.

Sout(arr[0]); }

| # In integers it pass by values.

Psvm change(int[] xc) {

| Values.

xc[0] = 90;

| Pass by Value

}.

| ↓ ↓ X → X

| Reference Value

| X X

| Not change in main Variable name.

⑥ Copy Of Array :-

⇒ Code :- Psvm {

```
int [] arr = { 60, 50, 40, 30, 20, 10 };
```

// Shallow Copy (Same array to new name dena)

```
int [] nums = arr;
```

nums[0] = 70; } → It will make changes in main array.

```
cout (arr[0]);
```

// Deep Copy (Creating New array with same elements).

// Method - 1

```
int [] brr = Arrays.copyOf(arr, arr.length);
```

brr[0] = 80;

```
cout (arr[0]);
```

cout (brr[0]);

No Change in main array.

It is built in method.
have to Import

// Method - 2.

```
int [] crr = new int [arr.length];
```

```
for (int i = 0; i < arr.length; i++) {
```

crr[i] = arr[i];

```
};
```

crr[0] = 100;

```
cout (arr[0]);
```

Without
using any
Build in
Method.

⑥ Basic Of ArrayList :-

M	T	W	T	F	S	S
Page No.:	YOUVA					
Date:						

→ Code :- PsVm {

```
ArrayList<Integer> arr = new ArrayList<>(); } use to create  
arr.add(0, 10); // arr[0] = 10. } New array list.  
arr.add(1, 20); // Initialize.  
arr.add(2, 30);  
arr.add(3, 40); }  
    ↗ Index ↗ Initialised this value.
```

cout(arr.get(0)); } → To print a particular index.

```
for (int i = 0; i < arr.size(); i++) { } for.  
    cout(arr.get(i) + " "); } Pointing of array  
} list.
```

3.

cout(arr); } → It will also work to print ArrayList.

```
arr.set(2, 300); } update / modify.  
cout(arr); } ↗ Index
```

arr.add(90); } → Use to add elements from back to ArrayList.

cout(arr.size()); } → Gives the size/Length of ArrayList.

```
for (int ele : arr) { } For Each Loop same as array.  
    cout(ele + " "); }  
}.
```

cout(arr);

3.

→ Why ArrayList

In Java, arrays they have fixed size.

So, we use dynamic array using OOP.

You can say that Unlimited Size Array.

→ To create ArrayList we need to import :-
import java.util.ArrayList; package on.
import java.util.List;

By importing this you can also write List<Integer> arr.....

⑦ LeetCode Question No. 1. [Two Sum]

M	T	W	T	F	S
Page No.	YOUVA				
Date					

Q. Find the doublet in the Array whose sum is equal to the given value "x".

→ Code :- Psym {

```

int[] arr = {2, 5, 3, 6, 1};
int x = 7;
for (int i = 0; i < arr.length; i++) {
    boolean flag = false; // optimization
    for (int j = i + 1; j < arr.length; j++) {
        if (arr[i] + arr[j] == x) {
            System.out.println(arr[i] + " + " + arr[j] + " = " + x);
            flag = true; // optimization
            break; // optimization
        }
    }
    if (flag == true) break; // optimization
}
}

```

→ Optimizations are to stop the Compilation whenever it gets satisfied output based on conditions.

arr = {2, 5, 3, 6, 1} x = 7;

Check [arr[i] + arr[j] == x]

⑧ Sort array of 0's & 1's.

arr = {1, 0, 0, 1, 1, 0}; n = arr.length.

[Method - 1] → Two - pass solution.

Step 1 → Calculate the No of 0's & 1's. ⇒ Using Two Pointer technique.
4 Ones by Using Loop. ⇒ i = 0, j = n - 1.

Step 2 → Store it in the variable Noz. ⇒ Create white Loop (i < j). {
No. of 0's & No. of 1's.

Step 3 → Create for loop from i = 0 to n
{ if (i < noof0ones) { arr[i] = 0; }
else
 arr[i] = 1;
}

if (arr[i] == 0) i++;
else if (arr[i] == 1) j++;
else if use swap technique
& j++, i++;
You will get the required result

⑧ Reverse Array.

Q. Write a Program to reverse the array without using any extra array.

⇒ Code :- Psvm {

```
int[] arr = { 10, 20, 30, 40, 50, 60, 70, 80 };  
int n = arr.length;
```

// Method - 1

```
for (int i = 0; i < n/2; i++) {
```

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

```
    arr[i] = arr[n-1-i];
```

```
    arr[n-1-i] = temp;
```

```
}
```

// Using Two Pointers.

```
int i = 0, j = n - 1;
```

// int i = 1, j = 5 ; Part of array reverse.

```
while (i <= j) {
```

```
# swap(arr, i, j);
```

```
i++;
```

```
j--;
```

```
}
```

};

⇒ Function:-

```
Ps v swap(int[] arr, int i, int j) {
```

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

```
    arr[i] = arr[j];
```

```
    arr[j] = temp;
```

```
}
```

arr = { 10, 20, 30, 40, 50, 60, 70, 80 }.

i + j = n - 1.

j = n - 1 - i,

→ i > j } Break the loop.

using temp.

arr[i]

arr[j]

using temp

You can do this using New array also.

⑨ Leet Code. [Question No. :-]

Q. Rotate the given array by k steps, where k is Non-Negative.

⇒ Code :- `psvm psvm {`

`int [] nums = { 10, 20, 30, 40, 50, 60, 70 };`

`int n = nums.length;`

`int k = 4;`

`k = k % n;`

`reverse (nums, 0, n-k-1);`

`reverse (nums, n-k, n-1);`

`reverse (nums, 0, n-1);`

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

`System.out.println (nums[i] + " ");`

`}`

`}`

Function / Method :-

`psvm reverse (int [] nums, int i, int j) {`

`while (i <= j) {`

`int temp = nums[i];`

`nums[i] = nums[j];`

`nums[j] = temp;`

`i++;`

`j++;`

`}`

`}`

so use

$K = K \% n$

$\{ \text{nums} = 10, 20, 30, 40 \quad \text{for } K=9 \}$

$\{ \text{nums} = 40, 10, 20, 30 \quad \text{for } n=4 \}$

$\{ \text{nums} = 40, 10, 20, 30 \}$

$\{ \text{nums} = 40, 10, 20, 30 \}$

Because every rotation it is going to be same.

1st Reverse this.

$\{ \text{nums} = 10, 20, 30, 40, 50, 60, 70 \}$

$K=3$

$n=7$

from 1st & 2nd $\Rightarrow 40, 30, 20, 10, 70, 60, 50.$

Then Reverse this whole you will get your result.

Final $\Rightarrow \{ 50, 60, 70, 10, 20, 30, 40 \}$.

M-1 \Rightarrow You can also solve it using ~~two-pass~~ Selection

(11) Leet Code [Question No. :-]

Solution	M	T	W	T	F	S	S
Page No.:							YOUVA

Q. Sort the array of 0's, 1's, 2's. [Dutch Flag Algorithm]

~~Method~~ Code:- PsV swap(int[] arr, int mid, int lo) {

int temp = arr[mid];

arr[mid] = arr[lo];

arr[lo] = temp;

}

\Rightarrow Code:- PsVm {

int[] arr = {0, 1, 2, 0, 1, 2, 1, 2, 0, 0};

int n = arr.length;

// Method - 2 [Dutch Flag Algorithm]

int mid = 0, hi = n-1, lo = 0;

while(mid <= hi) {

if (arr[mid] == 0) {

swap(arr, mid, lo);

lo++; mid++;

},

else if (arr[mid] == 1) mid++;

else {

swap(arr, mid, hi);

hi--;

},

},

for(int ele : arr) {

sout(ele + " ");

},

arr = {0, 1, 2, 0, 1, 2, 1, 2, 0, 0}.

Three Pointer Approach.

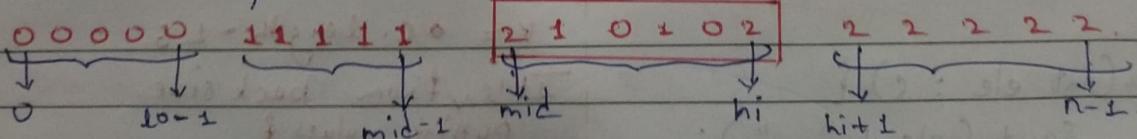
\rightarrow lo, mid, hi.

\Rightarrow Dutch Flag Algorithm.

\Rightarrow Break the Loop when mid > hi.

Steps: Break array in 4 parts.

lo
↑
unsorted part.



\Rightarrow lo se pehle
Sare 0 hone chayye.

\Rightarrow hi ke baad
Sare 2 hone
chayye.

\Rightarrow mid se pehle
Sare 1 hone
chayye.

(12) Leet Code Question No:-

M	T	W	T	R	S	S
Page No.:						YOUNA
Date:						

Q. Merge two sorted arrays in one single array.

⇒ Code:- Psvm {

```
int [] a = {11, 33, 42, 72};
```

```
int [] b = {26, 54, 69, 81};
```

```
int n = a.length + b.length;
```

```
int [] c = new int [n];
```

```
int i = 0, j = 0, k = 0;
```

```
while (i < a.length && j < b.length) {
```

```
if (a[i] <= b[j]) {
```

```
c[k] = a[i];
```

```
i++; k++;
```

? else / %

```
else {
```

```
c[k] = b[j];
```

```
j++; k++;
```

```
}
```

```
if (i == a.length) {
```

```
while (j < b.length) {
```

```
c[k] = b[j];
```

```
j++; i++; k++;
```

```
}
```

```
if (j == b.length) {
```

```
while (i < a.length) {
```

```
c[k] = a[i];
```

```
i++; k++;
```

```
}
```

```
for (int ele : c) {
```

```
cout < ele < " ";
```

```
}
```

3 - Pointer → i, j, k
 ↓ ↓ ↓
 (a) (b) (c)

You can do it in little different way also. same.

3 - Pointer only.

M-2,

i = m-1 ;

j = n-1 ;

k = o.c.length - 1 ;

It's from back side.
Putting values &

(23) Next Greatest Element.

Q. Next Greatest Element.

M	T	W	T	F	S	S
Page No.:						YOUVA
Date:						

⇒ Code :- Psvm {

```
int [] arr = { 12, 8, 60, 37, 2, 49, 46, 28, 21 };  
int n = arr.length;  
int [] ans = new int [n];  
ans [n-1] = -1;
```

// Method - 1 : Brute Force.

```
for (int i = 0; i < n-1; i++) {  
    int mx = Integer.MIN_VALUE;  
    for (int j = i+1; j < n; j++) {  
        mx = Math.max (mx, arr[j]);  
    }  
    ans [i] = mx;  
}
```

// Method - 2 : Optimised

```
int nge = arr [n-1];  
for (int i = n-2; i >= 0; i--) {  
    ans [i] = nge;  
    nge = Math.max (nge, arr [i]);  
}
```

```
for (int ele : arr) {  
    Sout (ele + " ");  
}
```

```
for (int ele : ans) {  
    Sout (ele + " ");  
}
```

3.

arr = { 12, 8, 60, 37, 2, 49, 46, 28, 21 };

$nge = \max (i, nge);$

~~ans~~ = { 60, 60, 49, 49, 49, 28, 28, 21, -1 }

$nge = \cancel{arr[n-1]}.$