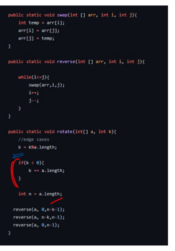


```
public static void swap(int [] arr, int i, int j){
    int temp = arr[i];
    arr[i] = arr[j];
    arr[j] = temp;
}

public static void reverse(int [] arr, int i, int j){

    while(i<=j){
        swap(arr,i,j);
        i++;
        j--;
    }
}

public static void rotate(int[] a, int k){
    //edge cases
    k = k%a.length;

    if(k < 0){
        k += a.length;
    }

    int n = a.length;

  reverse(a, 0,n-k-1);
  reverse(a, n-k,n-1);
  reverse(a, 0,n-1);
}
```

1 2 | 5 4 3

k=3

k = 3%5 = 3

k<0

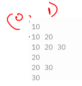
n-k-1 = 5-3-1 = 1

(0, 1)

n-k, n-1 → 5-3, 5-1 → (2, 4)

0, n-1 → 0, 5-1 → (0, 4)

34512



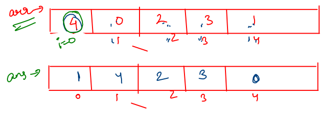
```
10
10  20
10  20  30
20
20  30
30
```

3 4 5 1 2

```
5
4
0
2
3
1
```

Sample Output

```
1
4
2
3
0
```



val = arr[i];
ans[val] = i;

val = arr[i];
newArr[val] = i; }

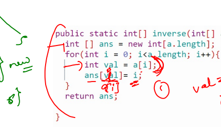
Inverse → i/p x

for

Question

1. Arr → I(Arr) } new

2. size = 7 { 0, 6 }



```
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;
}
```

4 0 2 3 1

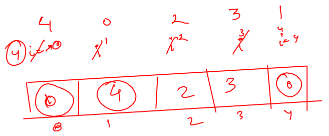
① val = a[0] = 4, i = 0

② val = 0

③ val = 2, ans(val) = ans(2) = 2

④ val = 3, ans(val) = ans(3) = 3

⑤ val = 1



Subarray → confuse. Subsequence, Sub set, substring

Subarray. arr [1 2 3 4 5]

continous segment

2 4 5 ×

3 4 5

2 3 4 ×, 4 3 2

1; 1 2; 1 2 3; 1 2 3 4; 1 2 3 4 5 → 5

2; 2 3; 2 3 4; 2 3 4 5 → 4

3; 3 4; 3 4 5 → 3

4; 4 5 → 2

5 → 1

5 1 2 ×

5+4+3+2+1 = 15

1+2+3+4+5 = n(n+1)/2

n → arr.length

5(5+1)/2 = 5×6/2 = 30/2 = 15

Subarray.

① → continous seq. (L → R)

② → unidirection

③ → n(n+1)/2 }

1 2 3 4 5

Subsequence. ① → It may or may not continous. unidirection (L → R)

②

③

2 4 5

1 2 3 4 5 →

Subset ⊃ Subarray

Subarray

10 20 30

10

10 20

10 20 30

20

20 30

30

aaa

a

10 20 30 → 2 2 2 = 8

Subsequence = 2^n

A (10)	B (20)	C (30)	
0	0	0	– – –
0	0	1	– – 30
0	1	0	– 20 –
0	1	1	– 20 30
1	0	0	10 – –
1	0	1	10 – 30
1	1	0	10 20 –
1	1	1	10 20 30

10; 10 20; 10 30; 10 20 30; 20; 20 30; 30

$2^n - 1$

8 ⇒ n=3

2^n 2^n = 2^3

Subarray Problem.

10

10 20

10 20 30

20

20 30

30

[10 | 20 | 30]

st=0, end; st=1, end; st=2, end

= n(n+1)/2 = 3(4)/2 = 6

st=0 [10 , 0; 10 20, 1; 10 20 30, 2

st=1 [20, 1; 20 30, 2

st=2 [30

st → end

0 → 0, 1, 2

1 → 1, 2

2 → 2

i → [i, n-1]

[st, end]

{ Subarray, Substring } | { Subsequence, Subset }

"abc" → bc

102030 → 2030

continuos

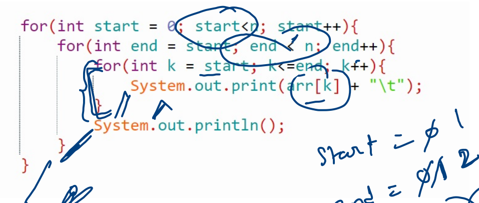
It may or may not

① Subarray →

② Subseq 2^n →

③

```
for(int start = 0; start<n; start++){
    for(int end = start; end < n; end++){
        for(int k = start; k<=end; k++){
            System.out.print(arr[k] + "\t");
        }
        System.out.println();
    }
}
```



10

10 20

20

n=2

2<2

Start = ~~0~~ ~~1~~ 2

end = ~~0~~ ~~1~~ 2

k=start = 0 1

① ② k = start = ~~0~~ 1

[0, 1]

0≤1

1≤1

k≤end

[10 | 20]

end < n

0<2

2(3)/2 = ③

2<2

k≤ end

0≤0

1≤ end

1≤0 ×

2<2

Start = ~~1~~ 2

end = ~~1~~ 2

k = ~~1~~ 2

k≤ end

1≤1

2≤1

k≤end.

10

10 20

20