ARRAY

Ques

Print in reverse

0	1	2	3	4	5
2	4	1	3	7	0

printReverse(arr, idx)

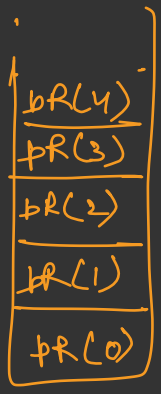
Expection	Faith	F E
0 7 3 1 4 2	0 7 3 1 4	pR(arr, 1);
pR(arr, 0)	pR(arr, 1)	Syso(arr[0]);

```
pR(arr, i+1);
Syso(arr[i]);
```

Main code

```
i == arr.length
return
```

Base case



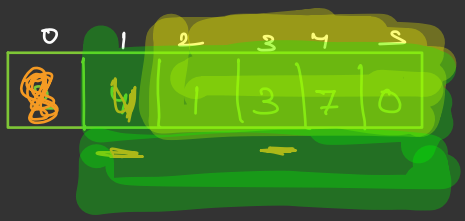
Ques Find Max

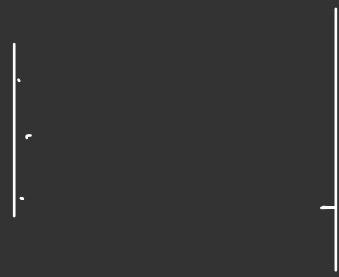
0	1	2	3	4	5
8	4	1	3	7	0

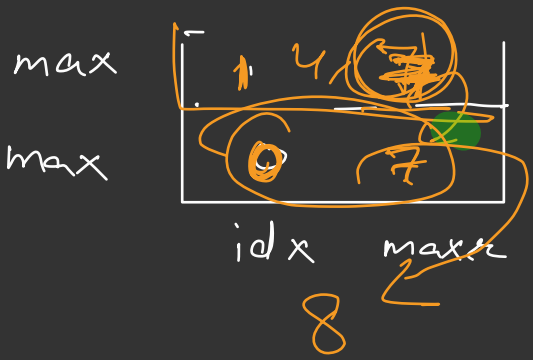
Max(arr, idx)

Expectation	Faith	F + E
max(arr, 0)	max(arr, 1)	int maxAns = max(arr,1)
8	7	int ans = maxAns > arr[i] ? maxAns : arr[i]

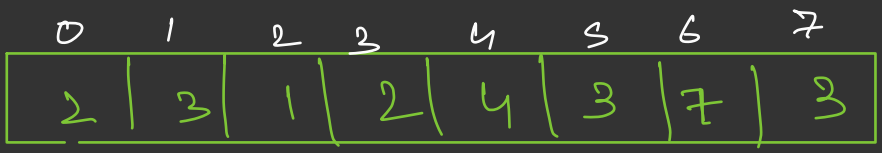
```
int max(int arr[], int i) {
    if( i == arr.length -1) return arr[i];
    int maxr = max(arr, i+1);
    return Math.max (maxr, arr[i]);
}
```







Ques Last Index



Last-Index (3)

⇒ 7

ϕ s int LI(int arr, int i, int v)

Exp

LI (arr, 0, 3)

⇒ 7

Faith

LI (arr, 1, 3)

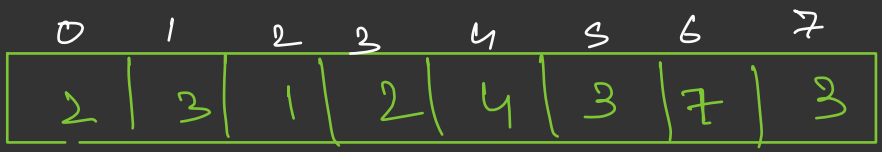
⇒ 7

If (arr != -1) return arr;
else if (arr[i] == val)
return i;

else return -1;

```
int LI(int arr[], int i, int v)
{
    int rr = LI(arr, i+1, v);
    If (rr != -1) return rr;
    else if (arr[i] == val)
        return i;
    else return -1;
}
```

Ques First Index



Expectation

FI(arr, 0, 2)

⇒ 0

Faith

FI(arr, 1, 2)

```
int FI(arr, i, v) {
    if (arr[i] == val)
        return i;
    return FI(arr, i+1, v);
```