

Product of Elements Except Itself

↓

α

0	2	1	0	4
0	1	2	3	4

product = $5 \times 2 \times 1 \times 2 \times 4 = 80$

$i=0, \text{ ans} = \frac{80}{5} = 16$

$i=1, \text{ ans} = \frac{80}{2} = 40$

$i=2, \text{ ans} = \frac{80}{1} = 80$

$i=3, \text{ ans} = \frac{80}{2} = 40$

$i=4, \text{ ans} = \frac{80}{4} = 20$

$O(N)$

arr

2	2	1	0	4
0	1	2	3	4

left

2	4	4	0	0
0	1	2	3	4

right

0	0	0	0	4
0	1	2	3	4

ans

0	0	0	16	0
0	1	2	3	4

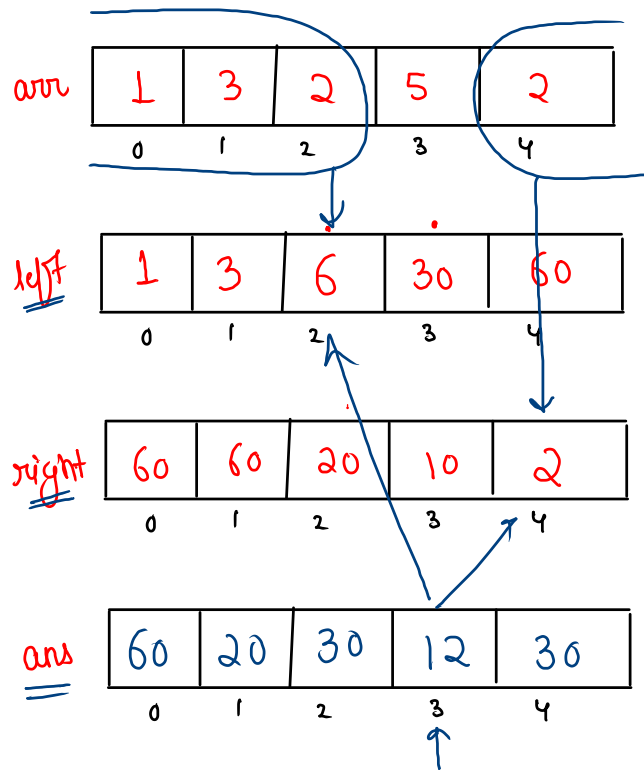
(product of all left elements including myself)

(product of all right elements including myself)

1) // product of all left elements including myself
 int[] left = new int[n];
 left[0] = arr[0];
 for (int i = 1; i < n; i++) {
 left[i] = left[i - 1] * arr[i];
 }

2) // product of all right elements including myself
 int[] right = new int[n];
 right[n - 1] = arr[n - 1];
 for (int i = n - 2; i >= 0; i--) {
 right[i] = right[i + 1] * arr[i];
 }

3) int[] ans = new int[n];
 ans[0] = right[1]; ←
 ans[n - 1] = left[n - 2]; ←
 for (int i = 1; i < n - 1; i++) {
 ans[i] = left[i - 1] * right[i + 1];
 }



code

```
public static void prod(int[] arr) {
    int n = arr.length;
```

```
    // product of all left elements including myself
    int[] left = new int[n];
    left[0] = arr[0];
    for (int i = 1; i < n; i++) {
        left[i] = left[i - 1] * arr[i];
    }
```

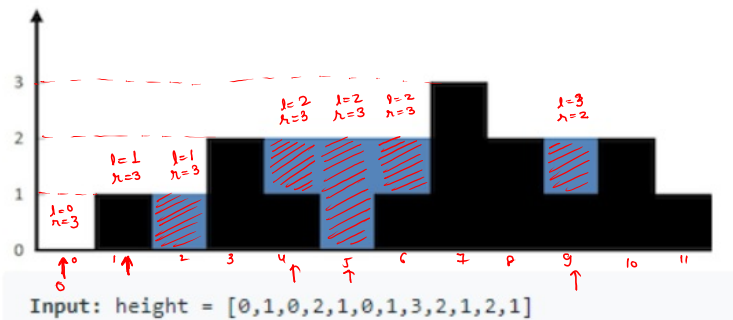
```
    // product of all right elements including myself
    int[] right = new int[n];
    right[n - 1] = arr[n - 1];
    for (int i = n - 2; i >= 0; i--) {
        right[i] = right[i + 1] * arr[i];
    }
```

```
    int[] ans = new int[n];
    ans[0] = right[1];
    ans[n - 1] = left[n - 2];
    for (int i = 1; i < n - 1; i++) {
        ans[i] = left[i - 1] * right[i + 1];
    }
```

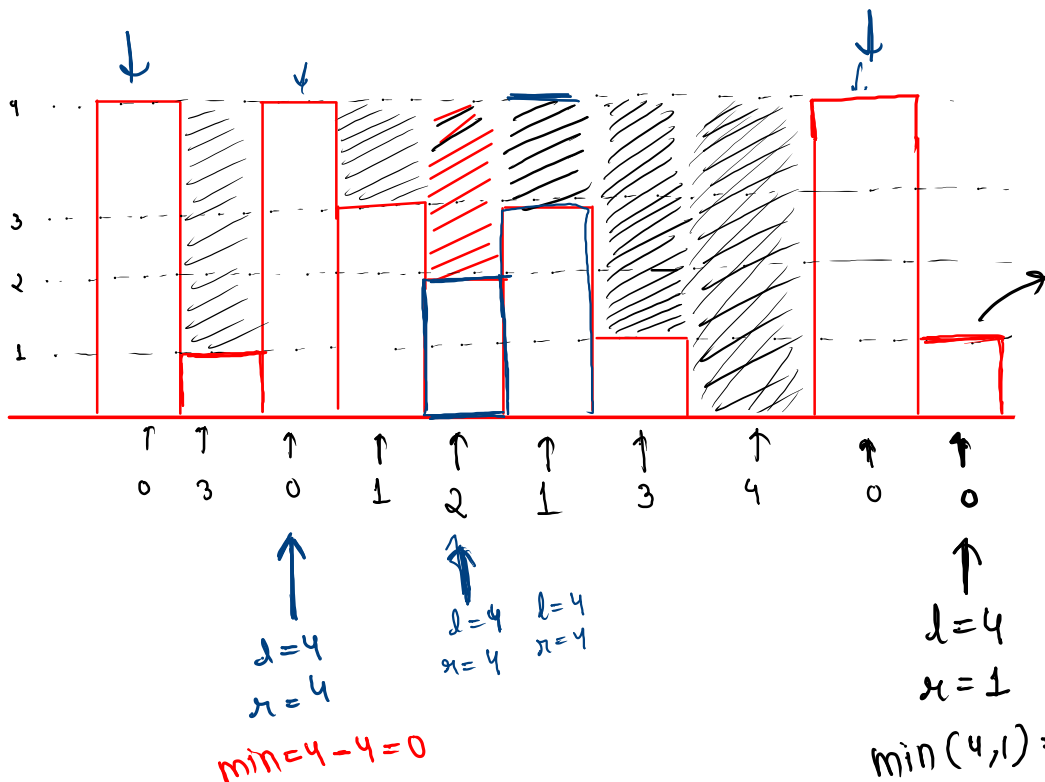
```
    // print answer array
    for (int i = 0; i < n; i++) {
        System.out.println( ans[i] );
    }
```

```
}
```

Ques Store maximum



6 units



logic:- 1) finding max height pillar on left and on right as well
2) min of left and right, exclude your pillar height

code

```
public static void storeWater(int[] arr, int n) {
    int water = 0;
```

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

```
        int left = arr[i];
        for (int j = 0; j < i; j++) {
            left = Math.max(left, arr[j]);
        }
```

```
        int right = arr[i];
        for (int j = i + 1; j < n; j++) {
            right = Math.max(right, arr[j]);
        }
```

```
        ans += Math.min(left, right) - arr[i];
    }
```

```
    System.out.println(water);
}
```