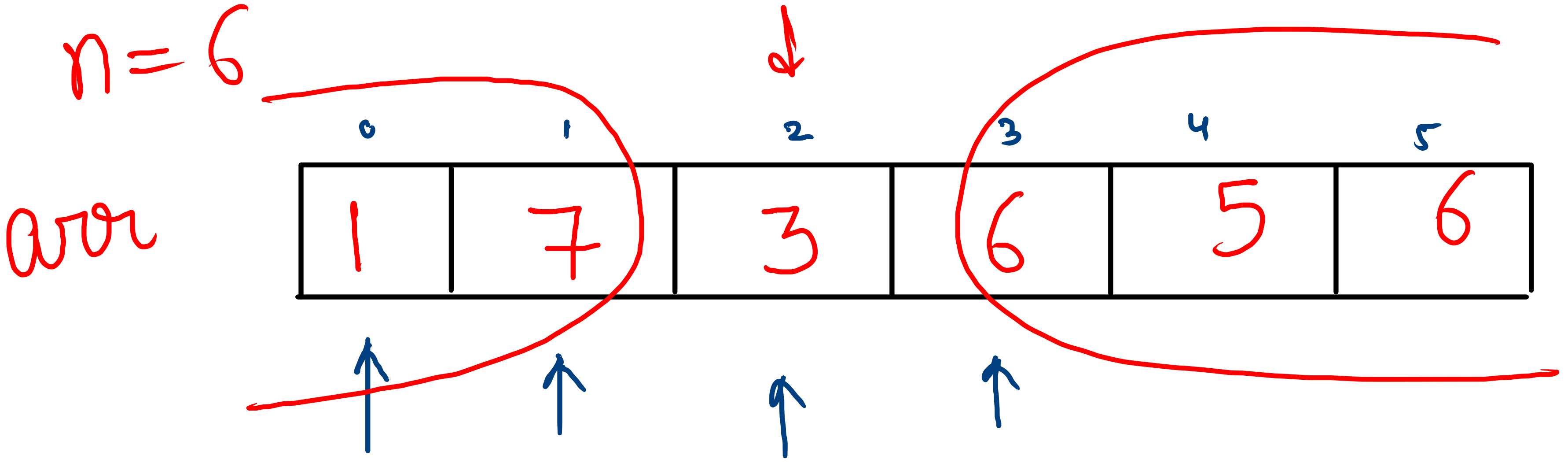
Find Pivot Index 1

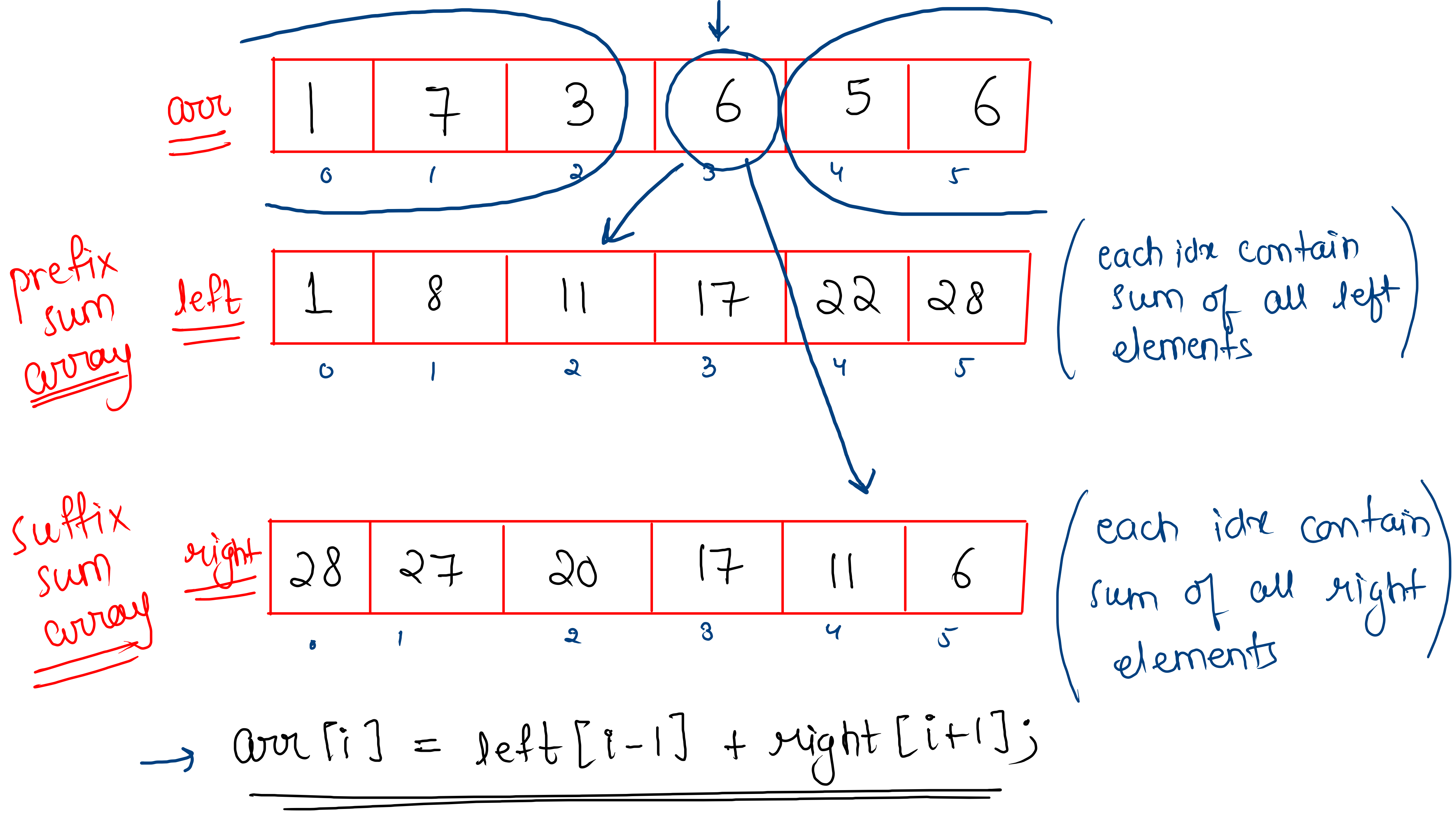
pivot :- where sum of all left ele. are equal to sum of all right ele.



left sum = ~~0~~ ~~1~~ ~~8~~ 11

right sum = ~~27~~ ~~20~~ ~~17~~ 11

pivot = 3



→ arr[i] = left[i-1] + right[i+1];

```java
public static int pivotIndex(int[] arr, int n) {
    int[] left = new int[n];
    left[0] = arr[0];
    for (int i = 1; i < n; i++) {
        left[i] = left[i - 1] + arr[i];
    }

    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];
    }

    if ( n > 0 && right[1] == 0 ) {
        return 0;
    }

    if ( n > 0 && left[n - 2] == 0 ) {
        return n - 1;
    }

    for (int i = 1; i < n - 1; i++) {
        if (left[i - 1] == right[i + 1]) {
            return i;
        }
    }
    return -1;
}
```

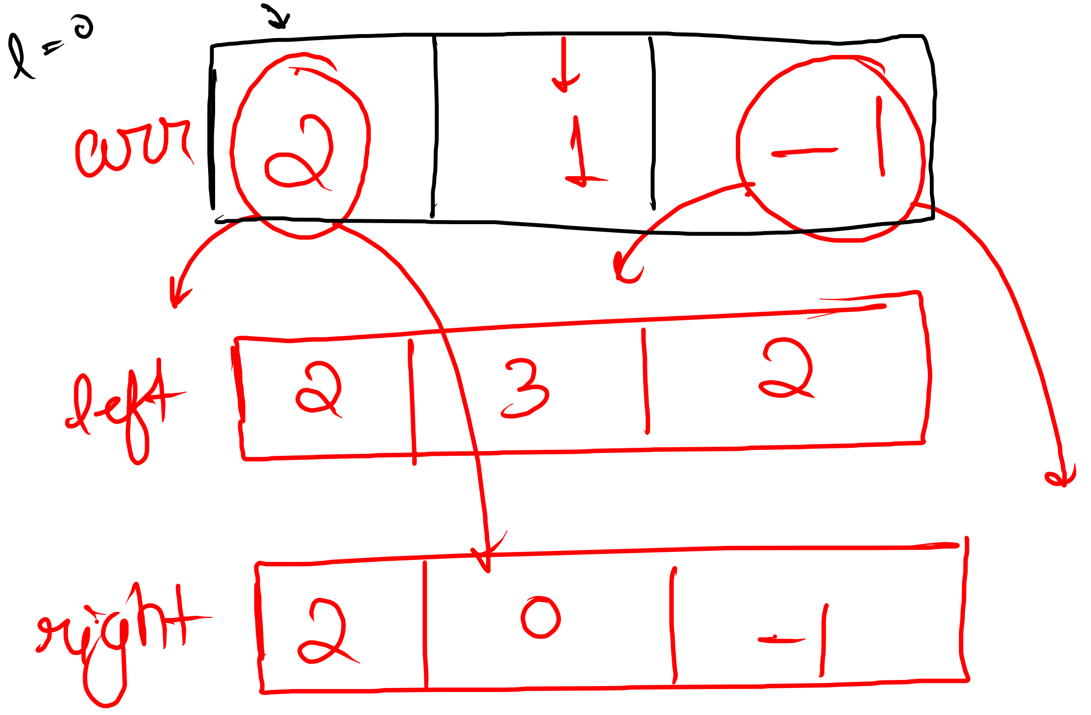
left

right

idx 0

idx n-1

logic



i=0, leftsum=0
rightsum= right[1]=0

i=2, leftsum=3
rightsum=0

Product of Elements Except Itself

$n=3$

	0	1	2
arr	2	5	3

$i=0, \quad 5 * 3 = 15$

$i=1, \quad 2 * 3 = 6$

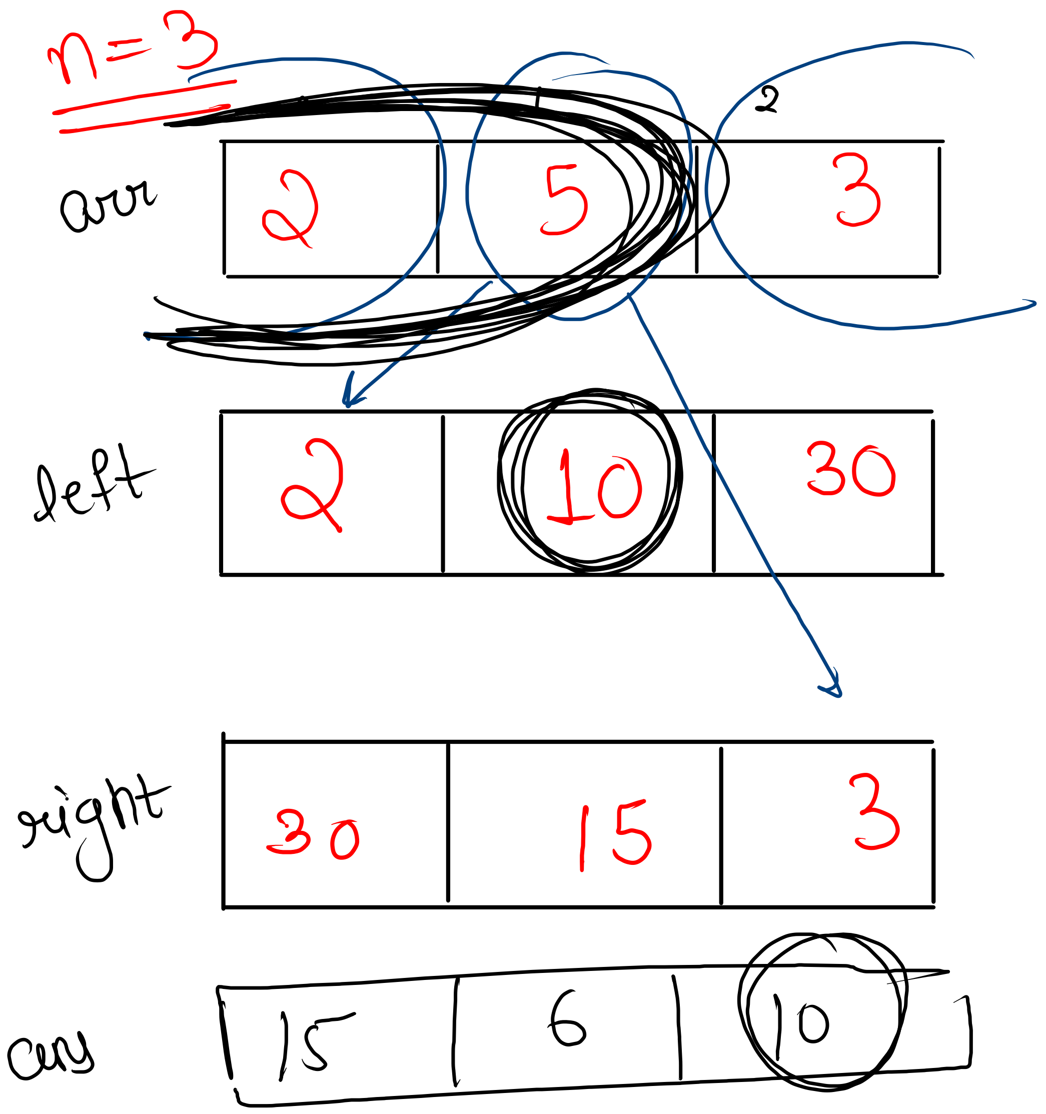
$i=2, \quad 2 * 5 = 10$

prefix product array

2	10	30

Suffix product array

30	15	3



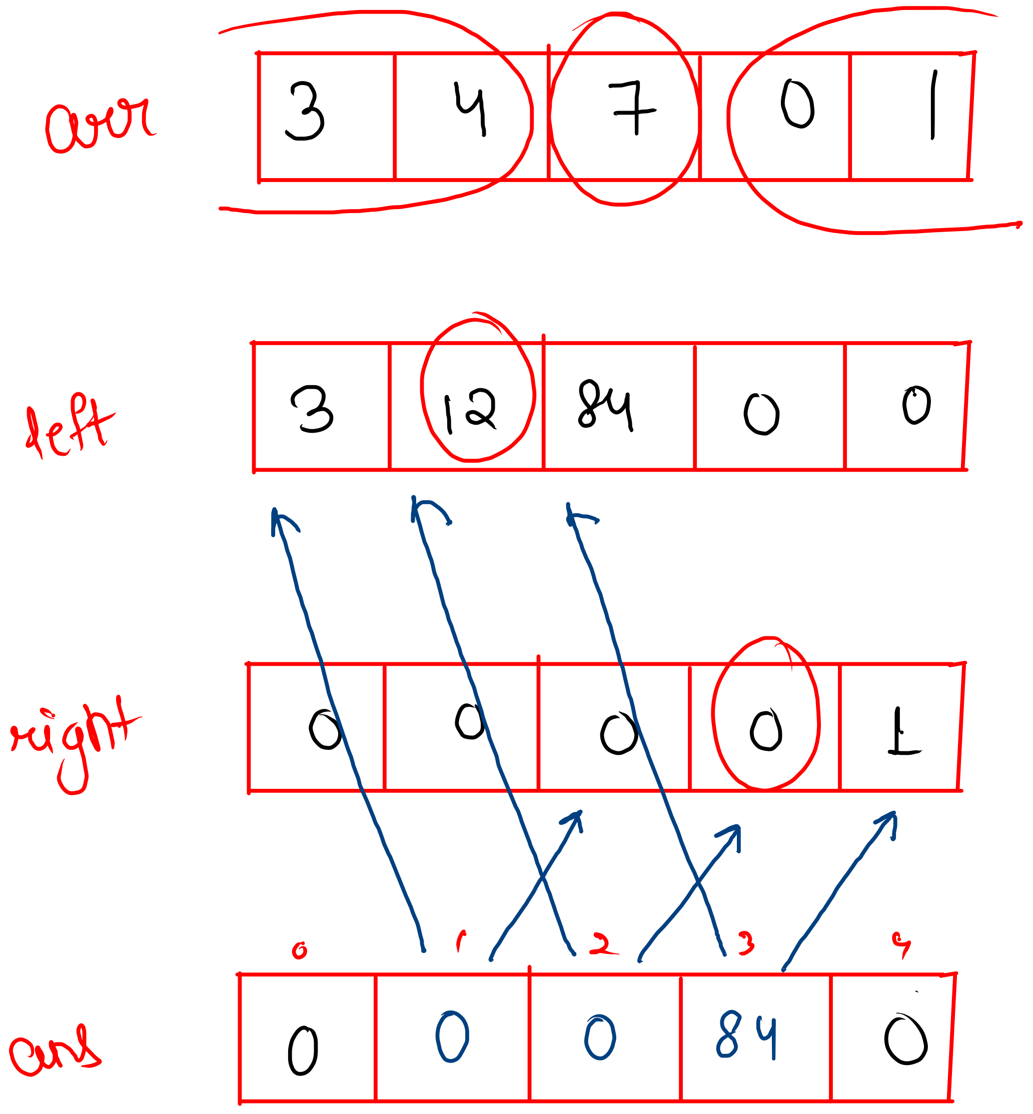
T. C = O(N)
S. C = O(N)
} n is size of array

```java
public static int[] productExceptItself(int[] arr, int n) {
    int[] left = new int[n];
    left[0] = arr[0];
    for (int i = 1; i < n; i++) {
        left[i] = left[i - 1] * arr[i];
    }

    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];
    }

    return ans;
}
```

```java
public static int[] productExceptItself(int[] arr, int n) {
    int[] left = new int[n];
    left[0] = arr[0];
    for (int i = 1; i < n; i++) {
        left[i] = left[i - 1] * arr[i];
    }

    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];
    }

    return ans;
}
```

$n=5$

1,2,3



arr
3
4
7
0
1
left
3
12
84
0
0
right
0
0
0
0
1
ans
0
1
2
3
4
0
0
0
84
0