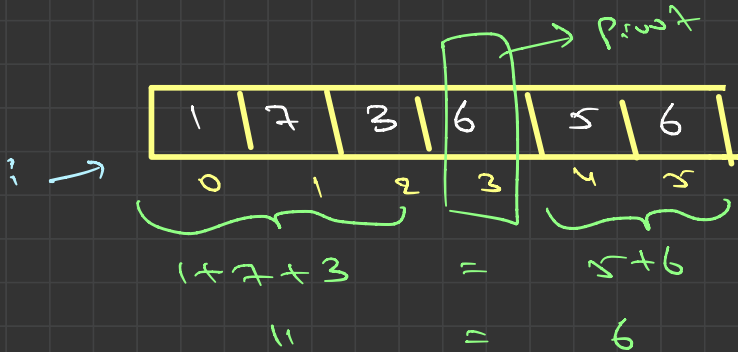



find pivot Index

→ 724

The sum of all the no. strictly to the left = sum of all the no. strictly to the right



return leftmost index

Exclude current index

Approach -1

Brute force

→ $TC = O(n)$
 $SC = O(1)$

① $low \rightarrow [0 \rightarrow n]$

② Calculate left sum before current index

③ Calculate right sum after current index

④ if both equal return true

⑤ else return false

1	7	3	6	5	6
0	1	2	3	4	5

$$i = 0$$

↓

1

$$lsum = 0$$

$$rsum = 7 + 3 + 6 + 5 + 6 \Rightarrow 27$$

$$lsum \neq rsum$$

$$i = 1$$

↓

7

$$lsum = 0 + 1 \Rightarrow 1$$

$$rsum = 0 + 3 + 6 + 5 + 6 \Rightarrow 20$$

$$lsum \neq rsum$$

$$i = 2$$

↓

3

$$lsum = 0 + 1 + 7 \Rightarrow 8$$

$$rsum = 0 + 6 + 5 + 6 \Rightarrow 18$$

$$lsum \neq rsum$$

$$i = 3$$

↓

6

$$lsum = 0 + 1 + 7 + 3 \Rightarrow 11$$

$$rsum = 0 + 5 + 6 \Rightarrow 11$$

$$lsum = rsum$$

↓
return true

Approach -2

Optimal Approach

initially 0

	0	1	2	3	4	5	
	1	7	3	6	5	6	← num
lsum =	0	1	8	11	18	23	
rsum	28	21	18	11	6	0	← initially 0

Exclude current index

Calculate sum from right

```
vector<int> lsum (num.size(), 0);  
vector<int> rsum (num.size(), 0);
```

```
for (int i = 1; i < num.size(); i++)  
{  
    lsum[i] = lsum[i-1] + num[i-1];  
}
```

```
for (int i = num.size() - 2; i > 0; i--)  
{  
    rsum[i] = rsum[i+1] + num[i+1];  
}
```

```
for (int i = 0; i < num.size(); i++)  
{  
    if (lsum[i] == rsum[i])  
        return true;  
}
```