

Count 1s in a Sorted Binary Array

[Courses](#)[Tutorials](#)[Jobs](#)[Practice](#)[Contests](#)

Given a binary array `arr[]` of size `N`, which is sorted in **non-decreasing order**, count number of **1's** in it.

Examples:

Input: `arr[] = {0,0, 0, 0, 0, 1, 1}`

Output: 2

Input: `arr[] = {1, 1, 1, 1, 1, 1, 1}`

Output: 7

Input: `arr[] = {0, 0, 0, 0, 0, 0, 0}`

Output: 0

Method: Binary Search

[C++](#)[Java](#)

```
#include <iostream>
using namespace std;

int countOnes(int arr[], int n)
{
    int low = 0, high = n - 1;

    while(low <= high)
    {
        int mid = (low + high) / 2;

        if(arr[mid] == 0)
            low = mid + 1;
        else
        {
            if(mid == 0 || arr[mid - 1] == 0)
```

Track Progress

21 of 60 Complete. (35%)

`high = mid - 1;`

Dash



All

```
return 0;  
}
```



Articles

```
int main() {
```



Videos

```
int arr[] = {0, 0, 1, 1, 1, 1}, n = 6;
```

```
cout << countOnes(arr, n);
```



Problems

```
return 0;  
}
```



Quiz



Contest

Output

4

Time complexity: $O(\log(N))$ **Auxiliary Space:** $O(1)$ [Mark as Read](#)[Report An Issue](#)

If you are facing any issue on this page. Please let us know.

[Menu](#)

Track Progress

21 of 60 Complete. (35%)