05 March 2025 12:01

1004. Max Consecutive Ones III

Solved

Medium Topics Companies Hint

Given a binary array `nums` and an integer `k`, return *the maximum number of consecutive* `1`*'s in the array if you can flip at most* `k` `0`'s.

Example 1:

> **Input:** nums = [1,1,1,0,0,0,1,1,1,1,0], k = 2
> **Output:** 6
> **Explanation:** [1,1,1,0,0,**1**,1,1,1,1,**1**]
> Bolded numbers were flipped from 0 to 1. The longest subarray is underlined.

Example 2:

> **Input:** nums = [0,0,1,1,0,0,1,1,1,0,1,1,0,0,0,1,1,1,1], k = 3
> **Output:** 10
> **Explanation:** [0,0,1,1,**1**,**1**,1,1,1,**1**,1,1,0,0,0,1,1,1,1]
> Bolded numbers were flipped from 0 to 1. The longest subarray is underlined.

Better approach :

```java
public int longestOnes(int[] nums, int k) {
    int cnt=0 , maxl=0 , l=0 , r=0 , n=nums.length;

    while(r<n){
        if(nums[r]==0) cnt++;
        while(cnt>k){
            if(nums[l]==0) cnt--;
            l++;
        }

        if(cnt<=k){
            maxl=Math.max(maxl , r-l+1);
        }
        r++;
    }
    return maxl;
}
```

Approach of question :
Question is like … find the length of longest subarray having the number of zero is k ….

Tc -o(n+n)

Optimal

```java
public int longestOnes(int[] nums, int k) {
    int cnt=0 , maxl=0 , l=0 , r=0 , n=nums.length;

    while(r<n){
        if(nums[r]==0) cnt++;
        if(cnt>k){
            if(nums[l]==0) cnt--;
            l++;
        }

        if(cnt<=k){
            maxl=Math.max(maxl , r-l+1);
        }
        r++;
    }
    return maxl;
}
```

Tc -o(n)

Brute force approach

```java
public int longestOnes(int[] nums, int k) {
    int maxl=0;
    int cnt=0;
    for(int i=0;i<nums.length;i++){

        for(int j=i;j<nums.length;j++){
            if(nums[j]==0) cnt++;
            if(cnt<=k)
            maxl=Math.max(maxl , j-i+1);
            else break;
        }
        cnt=0;
    }
    return maxl;
}
```

Tc -o(n2)