Remove duplicates in the array

BRUTE FORCE:

Push all elements into a hashset then iterate the hashset and initialise k=set.size() and copy the elemnts of set to the array and then return set.size() as ans

```
class Solution {
public:
    int removeDuplicates(vector<int>& nums) {
    set<int> s;
    for(int i=0;i<nums.size();i++)
    {
        s.insert(nums[i]);
    }
    int i=0;
    for(auto it: s)
    {
        nums[i++]=it;
    }
    return s.size();
    }
};
```

- Time Complexity : O(NlogN)
- Space Complexity : O(N)

Optimal Approach :

Using two pointer keep a pointer j at 0th position and traverse the array using loop wherever the value of array at current position and jth positions does not match we increment the value of j and set it to the differentiating element.

```
class Solution {
public:
    int removeDuplicates(vector<int>& nums) {
        int j=0;
        for(int i=0;i<nums.size();i++)
        {
            if(nums[i]!=nums[j])
            {
                j++;
                nums[j]=nums[i];
```

```
            }
        }
        return j+1;
    }
};
```

- Time Complexity : O(N)
- Space Complexity : O(1)