NAME-Vishavjeet Singh	UID-22BCS11878
BRANCH-BE-CSE	SECTION/GROUP-IOT-638-A
SEMESTER-6	DATEOFPERFORMANCE-12/02/2023
SUBJECTNAME-AP-LABII	SUBJECTCODE-22CSP-351

ASSIGNMENT

PROBLEM1:SortColors

Givenanarraynumswithnobjectscoloredred,white,orblue,sortthemin-placesothat objectsofthesamecolorareadjacent,withthecolorsintheorderred,white,andblue.We willusetheintegers0,1,and2torepresentthecolorred,white,andblue,respectively.You must solve this problem without using the library's sort function.

Example1:

Input:nums=[2,0,2,1,1,0]Output:[0,0,1,1,2,2]

Example2:

Input:nums=[2,0,1]Output:[0,1,2]

Constraints:

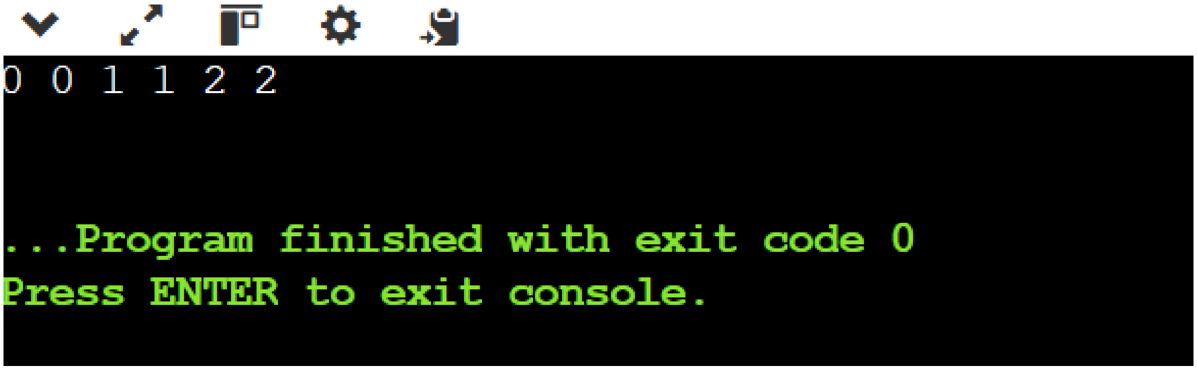
n==nums.length 1<=n<=300nums[i]iseither0, 1,or 2.

Followup:Couldyoucomeupwithaone-passalgorithmusingonlyconstantextraspace?

2.CODE

```
#include <iostream>
#include <vector>
usingnamespacestd;

classSolution{ public:
  voidsortColors(vector<int>&nums){
        intlow=0,mid=0,high=nums.size()-1; while
        (mid <= high) {
            if(nums[mid]==0){
                swap(nums[low], nums[mid]);
                low++;
                mid++;
            }elseif(nums[mid]==1){ mid++;
            }else{//nums[mid]==2
                swap(nums[mid], nums[high]);
                high--;
            }
        }
    }
};

intmain(){
    vector<int>nums={2,0,2,1,1,0};
    Solution sol;
    sol.sortColors(nums);

    for(intnum:nums){
        cout<<num<<"";
    }
    cout<<endl;
    return 0;
}
```

OUTPUT:



```
0 0 1 1 2 2



...Program finished with exit code 0
Press ENTER to exit console.
```

LEARNINGOUTCOMES:

1. **UnderstandingtheDutchNationalFlagAlgorithm**–Learnedhowtoefficientlysortanarray containingthreedistinctelements(0,1,2)in**onepass(O(n))**usinga**three-pointerapproach**.

2. **In-PlaceSortingWithoutExtraSpace**–Developedskillstosortthearray**withoutusingextra memory (O(1))**, making the solution space-efficient.

3.**EfficientArrayManipulationwithSwap Operations**–Gainedhands-onexperience in swappingelementsstrategicallyusing**low,mid,andhighpointers**toensurecorrectordering.

4.**OptimizingSortingWithoutUsingBuilt-inFunctions**–Learnedhowtomanuallyimplement sorting logic **without relying on sort()**, which is useful for interviews and competitive programming.

PROBLEM2:KthLargestElementinanArray

Given an integer array nums and an integer k, return the kth largest element in the array. Notethatitisthekthlargestelementinthesortedorder,notthekthdistinctelement.Can you solve it without sorting?

Example1:

Input:nums=[3,2,1,5,6,4],k=2Output: 5

Example2:

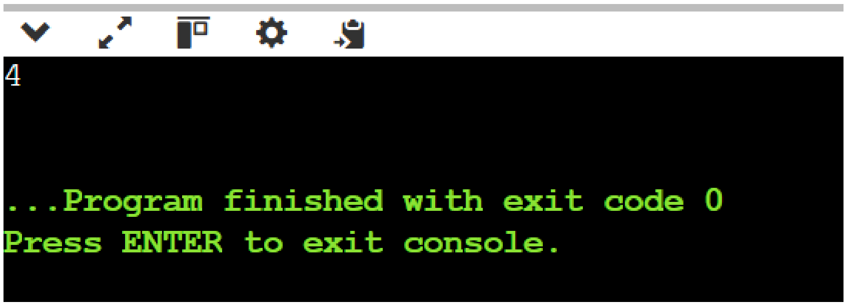
Input:nums=[3,2,3,1,2,4,5,5,6],k=4Output:4 Constraints:

1<=k<=nums.length<= 105-104<=nums[i]<=104

CODE:

```
#include <iostream>

#include <vector>

#include <queue>

usingnamespacestd;


classSolution{ public:

  intfindKthLargest(vector<int>&nums,intk){

    priority_queue<int,vector<int>,greater<int>>minHeap;


    for (int num : nums) {

      minHeap.push(num);if(

      minHeap.size()>k){

        minHeap.pop();//Removesmallestelementtomaintainsizek

      }

    }


    returnminHeap.top();//Therootofthemin-heapisthekthlargestelement

  }

};


intmain(){

  vector<int>nums={3,2,3,1,2,4,5,5,6}; intk=4;

  Solutionsol;

  cout << sol.findKthLargest(nums, k) << endl; // Output: 4 return

  0;
}
```

OUTPUT:



```
4


...Program finished with exit code 0
Press ENTER to exit console.
```

LEARNINGOUTCOMES:

1.**UnderstandingHeapDataStructure–**LearnedhowtouseaMin-Heap(PriorityQueue)to efficiently find the k-th largest element in O(n log k) time complexity.

2.**OptimizedSelectionWithoutSorting**–Developedtheabilitytofindthek-thlargestelement without sorting (O(n log n)), using a more efficient approach like Heap or Quickselect (O(n) average).

3.**EfficientSpaceUtilization**–GainedexperienceinsolvingproblemsusingO(k)extraspacefor the Min-Heap, making it memory-efficient compared to full sorting.

4.**ApplicationofQuickselectAlgorithm**–LearnedhowtoapplytheQuickselectAlgorithm(O(n) average case), a variation of QuickSort, to efficiently find the k-th largest element in an unorderedlist.