***Week – 8 (******06.06.2021 – 12.06.2021)***

***CODES BASED ON WEELKY TASK***

1. ***Subarrays with K Different Integers:***

class Solution {

public:

int subarraysWithKDistinct(vector<int>& A, int K) {

vector<int> m(A.size() + 1);

int i, j=0, prefix=0, cnt=0, res=0;

for(i=0; i<A.size(); i++)

{

if (m[A[i]]++ == 0) ++cnt;

if (cnt > K)

{

--m[A[j++]];

--cnt;

prefix = 0;

}

while (m[A[j]] > 1)

{

++prefix;

--m[A[j++]];

}

if (cnt == K) res += prefix + 1;

}

return res;

}

};

1. ***Sliding Window Median:***

class Solution {

public:

vector<double> medianSlidingWindow(vector<int>& nums, int k) {

vector<double> ans;

vector<double> v;

int i;

for(i=0; i<k-1; i++)

v.insert(lower\_bound(v.begin(),v.end(),nums[i]),nums[i]);

for(int i = k - 1;i < nums.size();i++)

{

v.insert(lower\_bound(v.begin(), v.end(), nums[i]), nums[i]);

if(k%2) ans.push\_back(double (v[k/2]));

else ans.push\_back(double (v[(k-2)/2] + v[k/2])/2);

v.erase(lower\_bound(v.begin(), v.end(), nums[i-k+1]));

}

return ans;

}

};

1. ***Grumpy Bookstore Owner:***

class Solution {

public:

int maxSatisfied(vector<int>& customers, vector<int>& grumpy, int minutes) {

vector<int> grumpyPos;

int maxCustCount = 0;

int currentCustCount = 0;

int i,j=0;

for(i=0; i<grumpy.size(); i++)

{

if (grumpy[i] == 1) grumpyPos.push\_back(i);

else currentCustCount += customers[i];

}

i = 0;

while (j < grumpyPos.size())

{

if (grumpyPos[j] - grumpyPos[i] + 1 <= minutes)

{

currentCustCount += customers[grumpyPos[j]];

j++;

}

else

{

if(currentCustCount > maxCustCount) maxCustCount=currentCustCount;

currentCustCount -= customers[grumpyPos[i]];

i++;

}

}

if(currentCustCount > maxCustCount) maxCustCount=currentCustCount;

return maxCustCount;

}

};