ARRAYS

SUBARRAY WITH GIVEN SUM

<https://practice.geeksforgeeks.org/problems/subarray-with-given-sum/0>

vector<int> subarraySum(int arr[], int n, int s){

int sum=0;

int l=0;

int i=0;

int flag=0;

for(;i<n;i++)

{

if(sum<s)

sum+=arr[i];

while(sum>s)

{sum-=arr[l];

l++;}

if(sum==s)

{

flag=1;

break;

}

}

vector<int>v;

if(flag==1)

{v.push\_back(l+1);

v.push\_back(i+1);}

else

{

v.push\_back(-1);

}

return v;

}

COUNTING THE TRIPLETS

<https://practice.geeksforgeeks.org/problems/count-the-triplets/0>

int countTriplet(int arr[], int n)

{

if(n<=2)

return 0;

sort(arr,arr+n);

int c=0;

for(int i=n-1;i>=0;i--)

{

int k=i-1;

int j=0;

while(j<k )

{

if(arr[i]==arr[j]+arr[k])

{c++;

k--;

j++;}

else if(arr[i]<arr[j]+arr[k])

k--;

else

j++;

}

}

return c;

}

KADANE’S THEOREM (MAXIMUM SUBARRAY SUM)

<https://practice.geeksforgeeks.org/problems/kadanes-algorithm/0>

int maxSubarraySum(int arr[], int n){

int maxi=INT\_MIN;

int sum=0;

for(int i=0;i<n;i++)

{

sum+=arr[i];

//cout<<sum<<" ";

if(sum>maxi)

maxi=sum;

if(sum<0)

sum=0;

}

return maxi;

}

MERGE WITHOUT EXTRA SPACE

<https://practice.geeksforgeeks.org/problems/merge-two-sorted-arrays/0/>

void merge(long long arr1[], long long arr2[], int n, int m)

{

for(int i=0;i<n;i++)

{

for(int j=0;j<m;j++)

{

if(arr1[i]>arr2[j])

{

swap(arr1[i],arr2[j]);

}

}

}

sort(arr1,arr1+n);

sort(arr2,arr2+m);

}

MISSING NUMBER

<https://practice.geeksforgeeks.org/problems/missing-number-in-array/0>

int MissingNumber(vector<int>& array, int n) {

int total=(n\*(n+1))/2;

for(int i=0;i<array.size();i++)

{

total-=array[i];

}

return total;

}

SORT AN ARAY OF 0,1 AND 2

<https://practice.geeksforgeeks.org/problems/sort-an-array-of-0s-1s-and-2s/0>

void sort012(int a[], int n)

{

int one=0,two=0,zero=0;

for(int i=0;i<n;i++)

{

if(a[i]==0)

zero++;

else if(a[i]==1)

one++;

else

two++;

}

//cout<<zero<<" "<<one<<" "<<two<<"\n";

int index=0;

while(zero>0)

{

a[index]=0;

zero--;

index++;

}

while(one>0)

{

a[index]=1;

one--;

index++;

}

while(two>0)

{

a[index]=2;

two--;

index++;

}

}

NUMBER OF PAIRS (x^y >y^x)

<https://practice.geeksforgeeks.org/problems/number-of-pairs/0/>

Logic-

For all y>x the condition holds true but there are some exceptions-

* If(x[i]==0) no pairs
* If(x[i]==1) no. of formed pairs=no. of 0’s in y

x smaller than y means x^y is greater than y^x. but some exceptions are-

* x = 2, y = 3 or 4
* x = 3, y = 2
* Else no of pair equal to all the y greater than x[i]+{cy[0]+cy[1]}-{(cy[4]+cy[3]) if x[i]==2}+{cy[2] if x[i]==3}

long long countPairs(int X[], int Y[], int M, int N)

{

long long int COUNTY[5]={0};

for(int i=0;i<N;i++)

{

if(Y[i]<5)

COUNTY[Y[i]]++;

}

sort(Y,Y+N);

long long int c=0;

for(int i=0;i<M;i++)

{

if(X[i]==1)

c+=COUNTY[0];

else

{

int\* idx = upper\_bound(Y, Y + N, X[i]);

int ans = (Y + N) - idx;

ans+= (COUNTY[0] + COUNTY[1]);

if (X[i] == 2)

ans -= (COUNTY[3] + COUNTY[4]);

if (X[i] == 3)

ans +=COUNTY[2];

c+=ans;

}

}

return c;

}

EQUILIBRIUM POINT

<https://practice.geeksforgeeks.org/problems/equilibrium-point/0>

class Solution{

public:

// Function to find equilibrium point in the array.

// a: input array

// n: size of array

int equilibriumPoint(long long a[], int n) {

int sum=0;

for(int i=0;i<n;i++)

{

sum+=a[i];

}

if(n==1)

return 1;

if(n==2)

return -1;

int sum1=0;

int flag=-1;

for(int i=0;i<n;i++)

{

sum1+=a[i];

if(sum1==sum)

{flag=i;

break;}

sum-=a[i];

}

if(flag==-1)

return -1;

return flag+1;

}

};

LEADERS IN AN ARRAY

<https://practice.geeksforgeeks.org/problems/leaders-in-an-array/0>

class Solution{

//Function to find the leaders in the array.

public:

vector<int> leaders(int a[], int n){

vector<int>v;

v.push\_back(a[n-1]);

int maxi=a[n-1];

for(int i=n-2;i>=0;i--)

{

if(maxi<=a[i])

v.push\_back(a[i]);

maxi=max(maxi,a[i]);

}

reverse(v.begin(),v.end());

return v;

}

};

MINIMUM PLATFORMS

<https://practice.geeksforgeeks.org/problems/minimum-platforms/0>

int findPlatform(int arr[], int dep[], int n)

{

sort(arr,arr+n);

sort(dep,dep+n);

int res=1,plat=1;

int i=1,j=0;

while(i<n && j<n)

{

if(arr[i]<=dep[j])

{

plat++;

i++;

}

else

{

plat--;

j++;

}

res=max(res,plat);

}

return res;

}

};

REVERSE IN GROUPS

<https://practice.geeksforgeeks.org/problems/reverse-array-in-groups/0>

void reverseInGroups(vector<long long>& arr, int n, int k){

int i=0;

while(i<n)

{

if(i+k<n)

{

int x=i;

int y=i+k-1;

while(x<y)

{

swap(arr[x],arr[y]);

x++;

y--;

}

}

else

{

int x=i;

int y=n-1;

while(x<y)

{

swap(arr[x],arr[y]);

x++;

y--;

}

}

i+=k;

}

}

K-th SMALLEST ELEMENT

<https://practice.geeksforgeeks.org/problems/kth-smallest-element/0>