VECTORS

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;

}