

Data Structure & Algorithm Lab II

Assignment 1

Trimester: Fall 2021

Course Code: CSE 2218

**Submitted To**

**Akib Zaman**

Lecturer

**Submitted By**

Sunny Sutradhar

ID: 011201084

Section: E

**Solution For question 1:**

**--Find Maximum Sub array sum**

#include<bits/stdc++.h>

using namespace std;

int maxCrossSum(int a[],int l,int mid,int h){

int sum = 0;

int leftSum = INT\_MIN;

int rightSum = INT\_MIN;

for(int i=mid;i>=l;i--){

sum+=a[i];

if(sum>leftSum)leftSum = sum;

}

sum=0;

for(int i=mid+1;i<=h;i++){

sum+=a[i];

if(sum>rightSum)rightSum=sum;

}

int totalSum = leftSum+rightSum;

return max(totalSum,max(leftSum,rightSum));

}

int maxSubArraySum(int a[],int l,int h){

if(l==h){

return a[l];

}

else{

int mid = (l+h)/2;

int leftSum = maxSubArraySum(a,l,mid);

int rightSum = maxSubArraySum(a,mid+1,h);

int crossSum = maxCrossSum(a,l,mid,h);

return max(crossSum,max(leftSum,rightSum));

}

}

int main(){

int n;

cin>>n;

int a[n];

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

cin>>a[i];

}

cout<<maxSubArraySum(a,0,n-1);

}

**Solution for Question 2:**

**--Quick Sort**

#include<bits/stdc++.h>

using namespace std;

int position(int a[],int l,int h){

int pivot = a[l];

int i = l+1;

int j = h;

while(i<j){

while(a[i]<pivot){

i++;

}

while(a[j]>pivot){

j--;

}

if(i<j)swap(a[i],a[j]);

}

swap(a[j],a[l]);

return j;

}

void quickSort(int a[],int l,int h){

if(l<h){

int pivot = position(a,l,h);

quickSort(a,l,pivot-1);

quickSort(a,pivot+1,h);

}

}

int main(){

int n;

cin>>n;

int a[n];

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

cin>>a[i];

}

quickSort(a,0,n-1);

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

cout<<a[i]<<" " ;

}

}

**Solution for Question 3:**

**--Kth largest element**

#include<bits/stdc++.h>

using namespace std;

//Time complexity Big Oh(nLog(n))

int kthLargestElement(int a[],int n,int k){

sort(a,a+n,greater<int>());

return a[k-1];

}

int main(){

int n;

cin>>n;

int a[n];

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

cin>>a[i];

}

int k;

cin>>k;

cout<<kthLargestElement(a,n,k)<<endl;

}

**Solution for question 4:**

**--Reverse Pair**

#include<bits/stdc++.h>

using namespace std;

typedef struct LinkedList node;

void marge(int a[],int l,int mid,int h){

int temp[h-l+1];

int i = l;

int j = mid+1;

int index=0;

while(i<=mid && j<=h){

if(a[i]<a[j]){

temp[index++] = a[i++];

}

else {

temp[index++] = a[j++];

}

}

while(i<=mid)

temp[index++] = a[i++];

while(j<=h)

temp[index++]=a[j++];

index = 0;

for(int i=l;i<=h;i++){

a[i] = temp[index++];

}

// cout<<endl;

}

void divideAndConqure(int a[],int l,int h,int ans[]){

if(l<h){

int mid = (l+h)/2;

divideAndConqure(a,l,mid,ans);

divideAndConqure(a,mid+1,h,ans);

int i=l;

int j=mid+1;

int counter = 0;

while(i<=mid){

if(j<=h && (a[i]>2\*a[j])){

counter++;

j++;

}else{

ans[0] += counter;

// cout<<ans[0]<<endl;

//counter = 0;

i++;

}

}

marge(a,l,mid,h);

}

}

int reversePairs(int a[],int n){

int ans[1]={0};

divideAndConqure(a,0,n-1,ans);

return ans[0];

}

int main(){

int n;

cin>>n;

int a[n];

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

cin>>a[i];

}

cout<<reversePairs(a,n)<<endl;

}