

```
#include<iostream>
using namespace std;

long long merge(int A[],int left,int mid,int right){

    int i=left,j=mid,k=0;

    int temp[right-left+1];
    long long count = 0;
    while(i<mid && j<=right){
        if(A[i] <= A[j]){
            temp[k++] = A[i++];
        }else{
            temp[k++] = A[j++];
            count += mid - i;
        }
    }
    while(i<mid){
        temp[k++] = A[i++];
    }
    while(j<=right){
        temp[k++] = A[j++];
    }

    for(int i=left,k=0;i<=right;i++,k++){
        A[i] = temp[k];
    }
    return count;
}

long long merge_sort(int A[],int left,int right){

    long long count = 0;
    if(right > left){
        int mid = (left + right)/2;

        long long countLeft = merge_sort(A,left,mid);
        long long countRight = merge_sort(A,mid+1,right);
        long long myCount = merge(A,left,mid+1,right);

        return myCount + countLeft + countRight;
    }
    return count;
}

long long solve(int A[], int n)
{
    long long ans = merge_sort(A,0,n-1);
```

```
    return ans;
}

int main(){
    int A[] = {5,4,2,3,1};
    cout << solve(A,5);
    return 0;
}
```