Day 82/180 Merge Sort

1: Sort an array in non-increasing order using Merge Sort.
2: Count Inversions

My Merge sort Code:

```
void merge(int arr[], int start, int mid, int end)
{
   vector<int>temp(end-start+1);
   int left = start, right = mid+1, index = 0;

   while(left<=mid&&right<=end)
   {
       if(arr[left]<=arr[right])
       {
           temp[index]=arr[left];
           index++, left++;
       }
       else
       {
           temp[index]=arr[right];
           index++, right++;
       }

   }

   // left array is not empty yet
   while(left<=mid)
   {
       temp[index]=arr[left];
       index++, left++;
   }
   // right array is not empty yet
   while(right<=end)
   {
        temp[index]=arr[right];
        index++, right++;
   }

   index=0;
  // put these value in array
  while(start<=end)
  {
   arr[start]=temp[index];
   start++, index++;
  }



}

void mergesort(int arr[], int start, int end)
{
   if(start==end)
   return;

   int mid = start+(end-start)/2;
   // left side
   mergesort(arr,start,mid);
   // right side
   mergesort(arr,mid+1,end);
   merge(arr,start,mid,end);
}



int main()
{
   int arr[] = {6,3,1,2,8,9,10,7,3,10};
   mergesort(arr,0,9);
   for(int i=0;i<10;i++)
   cout<<arr[i]<<" ";

}
```