归并排序
复杂度O(nlogn)
递归实现，可用于实现求交换相邻数字使得数组有序的最少次数。

```
#include <bits/stdc++.h>
//#define LOCAL_JUDGE
using namespace std;
typedef long long int ll;
void Merge(int *A,int *L,int left,int *R,int right)//合并操作
{
    int i,j,k;
    i=0;j=0;k=0;
    while(i<left && j<right){
        if(L[i]<R[j]){
            A[k++]=L[i++];
        } else {
            A[k++]=R[j++];
        }
    }
    while(i<left){
        A[k++]=L[i++];
    }
    while(j<right){
        A[k++]=R[j++];
    }
}
void Mergesort(int *arr,int n)//等分递归
{
    int mid,*L,*R;
    if(n<2) return;
    mid=n/2;
    L=new int[mid];
    R=new int[n-mid];
    for(int i=0;i<mid;i++) L[i]=arr[i];
    for(int i=mid;i<n;i++) R[i-mid]=arr[i];
    Mergesort(L,mid);
    Mergesort(R,n-mid);
    Merge(arr,L,mid,R,n-mid);
    delete [] R;
    delete [] L;
}

int num[100],temp[100];
int main()
{
#ifdef LOCAL_JUDGE
    freopen("Text.txt","r",stdin);
#endif // LOCAL_JUDGE
```

```
    int n;
    scanf("%d",&n);
    for(int i=0;i<n;i++)
    {
        scanf("%d",&num[i]);
    }
    Mergesort(num,n);
#ifdef LOCAL_JUDGE
    fclose(stdin);
#endif // LOCAL_JUDGE
    return 0;
}
```