**分治算法：把一个无序的数组，变成一个有序的数组**

**思想：**

**1.分而治之，将一个无序的数列一直一分为二，直到分到序列中只有一个数的时候，这个序列肯定是有序的，因为只有一个数，然后将两个只含有一个数字的序列合并为含有两个数字的有序序列，这样一直进行下去，最后就变成了一个大的有序数列**

**2.递归的结束条件是分到最小的序列只有一个数字的时候**

**时间复杂度分析：**

**最坏情况：T(n)=O(n\*lg n)**

**平均情况：T(n)=O(n\*lg n)**

**稳定性：稳定（两个数相等的情况，不用移动位置**

**辅助空间：O(n)**

**实验代码：**

#include<stdio.h>

#define max 1024

int SR[max],TR[max];

int merge(int SR[],int TR[],int s,int m,int t)

int i=s,j=m+1,k=s;

while(i<=m&&j<=t)

{

if(SR[i]<SR[j])

{

TR[k++]=SR[i++];

}else

{

TR[k++]=SR[j++];

}

}

while(i<=m)

{

TR[k++]=SR[i++];

}

while(j<=t)

{

TR[k++]=SR[j++];

}

return 0;

}

void copy(int SR[],int TR[],int s,int t)

{

int i;

for(i=s;i<=t;i++)

{

SR[i]=TR[i];

}

}

int mergesort(int SR[],int s,int t)

{

if(s<t)

{

int m=(s+t)/2;

mergesort(SR,s,m);

mergesort(SR,m+1,t);

merge(SR,TR,s,m,t)；

copy(SR,TR,s,t);

}else

{

}

return 0;

}

int main()

{

int n;

printf("请输入排序数字的个数:\n");

scanf("%d",&n);

int i;

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

{

scanf("%d",&SR[i]);

}

mergesort(SR,0,n-1);

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

{

printf("%d ",SR[i]);

}

printf("\n");

return 0;

}