**《数据结构》课程设计题目13：基数排序**

**2015级信息安全 赵于嫣 320150940740**

[题目内容]：A为每个关键字不超过3位的十进制整数关键字集合，试编写一个采用静态链表组织模式实现的基数排序程序将A进行由小到大的排序。

**[源代码]**

#include<stdio.h>

#include <stdlib.h>

#define N 1024

int RadixCountSort(int\* npIndex, int nMax, int\* npData, int nLen)

{

int\* pnCount=(int\*)malloc(sizeof(int)\* nMax); //保存计数的个数

int i=0;

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

{

pnCount[i]=0;

}

for (i=0;i<nLen;++i) //初始化计数个数

{

++pnCount[npIndex[i]];

}

for (i=1; i<10;++i) //确定不大于该位置的个数。

{

pnCount[i] +=pnCount[i-1];

}

int \* pnSort =(int\*)malloc(sizeof(int) \* nLen); //存放零时的排序结果。

//注意：这里i是从nLen-1到0的顺序排序的，是为了使排序稳定。

for (i=nLen-1;i>=0;--i)

{

--pnCount[npIndex[i]];

pnSort[pnCount[npIndex[i]]]=npData[i];

}

for (i=0;i<nLen;++i) //把排序结构输入到返回的数据中。

{

npData[i] = pnSort[i];

}

free(pnSort); //释放

free(pnCount);

return 1;

}

//基数排序

int RadixSort(int\* nPData, int nLen)

{

//申请存放基数的空间

int \*nDataRadix=(int \*)malloc(sizeof(int) \*nLen);

int nRadixBase=1; //初始化倍数基数为1

int nIsOk=0; //设置完成排序为false

int i;

//循环，知道排序完成

while (nIsOk==0)

{

nIsOk=1;

nRadixBase \*=10;

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

{

nDataRadix[i] = nPData[i] % nRadixBase;

nDataRadix[i] /= nRadixBase/10;

if (nDataRadix[i]>0)

{

nIsOk=0;

}

}

if (nIsOk==1) //如果所有的基数都为0，认为排序完成，就是已经判断到最高位了。

{

break;

}

RadixCountSort(nDataRadix, 10, nPData, nLen);

}

free(nDataRadix);

return 1;

}

int main()

{

int i=0;

int j;

int nData[N];

printf("----------------------基数排序-------------------------\n\n");

printf("请输入你要排序的个数：");

scanf("%d",&j);

if(j==0) return 0;

printf("请输入的%d个整数：\n",j);

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

{

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

}

RadixSort(nData, j);

printf("基数排序法排序后：\n");

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

{

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

}

printf("\n");

return 0;

}

**[运行结果]**

