

更多资料，微软公司等数据结构+算法面试 100 题见：

<http://topic.csdn.net/u/20101023/20/5652ccd7-d510-4c10-9671-307a56006e6d.html>

July、2010/11/01。

以下是，5 大内部排序算法性能的比较，之程序实现。

已编译通过。

```
-----
#include<stdio.h>
#include<stdlib.h>

#define N 8 // 用来存排序用的数据，其中第一个元素经常作为“哨兵”和交换用
int change[5],compare[5];
//分别用来描述排序所交换的趟数和比较的趟数从第二个 元素开始使用
int p=0;

void insretsor(int s[])
{
    int i,j;int sum=0;
    int a[8];
    for(i=1;i<N;i++)
    {
        a[i]=s[i];
    }
    for (i=2;i<N;i++)
    {
        a[0]=a[i];
        for(j=i-1;a[0]<a[j];)
        {
            compare[0]++;
            a[j+1]=a[j];
            change[0]++;
            j--;
        }
        a[j+1]=a[0];
        change[0]++;
        sum++;
        printf("第%d 排序结果是: ",sum);
        for(int i=1;i<N;i++)
        {
            printf("%5d",a[i]);
        }
        printf("\n");
    }
}
```

```

    }
    printf("一共进行了%d 比较",compare[0]);
    printf("一共进行了%d 交换",change[0]);
}

```

```

void bubblesort(int s[])
{
    int i=1,j,k,m,done=1;
    int a[8];
    for(m=1;m<N;m++)
    {
        a[m]=s[m];
    }
    while(i<N&&done)
    {
        done=0;
        for(j=1;j<N-i;j++)
        {
            if(a[j+1]<a[j])
            {
                a[0]=a[j];
                a[j]=a[j+1];
                a[j+1]=a[0];
                done=1;
                change[1]=change[1]+3;
            }
        }
        printf("第%d 趟的结果是: ",i);
        for(k=1;k<N;k++)
        {
            printf("%5d",a[k]);
        }
        printf("\n");
        compare[1]++;
        i++;
    }
}

```

```

printf("冒泡排序一共经过了%d 交换: ",compare[1]);
printf("冒泡排序一共经过了%d 比较: ",change[1]);
printf("最终排序结果是: \n");
for(k=1;k<N;k++)
    printf("%5d",a[k] );

```

```

        printf("\n");
    }

void simplessort(int s[])
{
    int i,j,k,sum=0;
    int a[8];
    for(i=1;i<N;i++)
    {
        a[i]=s[i];
    }
    for(i=1;i<=N-1;i++)
    {
        k=i;
        for(j=i+1;j<N;j++)
        {
            if(a[j]<a[k])k=j;compare[2]++;
        }
        if(k!=i)
        {
            a[0]=a[k];
            a[k]=a[i];
            a[i]=a[0];
        }
        change[1]=change[1]+3;
        sum++;
        printf("输出第%d 趟排序的结果: ",sum);
        for(int i=1;i<N;i++)
        {
            printf("%6d",a[i]);
            printf("\n");
        }
    }

    printf("一共进行了%d 比较:",compare[2]);
    printf("一共进行了%d 交换:",change[2]);
}

```

```

void q_sort(int s[],int left,int right)
{
    int i=left,j=right;
    int k;
    if(i<j)
    {

```

```

s[0]=s[i];
do{
    while(i<j && s[j]>=s[0])
        j--;
    if(i<j)
    {
        s[i]=s[j];
        compare[3]++;
        i++;
    }
    while(i<j && s[i]<=s[0])
        i++;
    if(i<j)
    {
        s[j]=s[i];
        compare[3]++;
        j--;
    }
    p++;
    printf("第%d 趟的结果是: ",p);
    for(int c=1;c<N;c++)
    {
        printf("%5d",s[c]);
    }
    printf("\n");
    change[3]++;
}
while(i<j);
s[i]=s[0];
printf("\n");
q_sort(s,left,j-1);
q_sort(s,j+1,right);
}
}

```

```

void merges(int s[],int s1[],int m,int n,int h)// 一次归并
{
    int i,j,k,t;
    k=i+m;
    j=n+1;

```

```

while(i<=n && j<=h)
{
    if(s[i]<=s[j])
    {
        s1[k]=s[i];
        i++;
    }
    else
    {
        s1[k]=s[j];
        j++;
    }
    k++;
}
if(i<=m)
{
    for(t=i;t<=m;t++)
        s1[k+t-i]=s[t];
}
else
    for(t=j;t<=h;t++)
        s1[k+t-j]=s[t];
}

```

```

void mergepass(int s[N],int s1[N],int len)
//一趟归并
{
    int i,j;
    i=1;
    while(i<=N-2*len+1)
    {
        merges(s,s1,i,i+len-1,i+2*len-1);
        i=i+2*len;
    }
    if(i+len-1<N)
        merges(s,s1,i,i+len-1,N-1);
    else
        for(j=i;j<N;j++)
            s1[j]=s[j];
}

```

```

void mergesort(int s[N])
{
    int a[8];
    int g;
    for(g=1;g<N;g++)
    {
        a[g]=s[g];
    }
    int len;
    int s1[8];
    len=1;
    while(len<N-1)
    {
        mergepass(a,s1,len);
        len=2*len;
        p++;
        printf("输出归并排序第%d 的结果:\n",p);
        for(int l=1;l<N;l++)
        {
            printf("%6d",s1[l]);
        }
        printf("\n");
        change[4]++;
        mergepass(s1,a,len);
        len=2*len;
        p++;
        printf("输出归并排序第%d 的结果:\n",p);
        for(g=1;g<N;g++)
        {
            printf("%6d",a[g]);
        }
        printf("\n");
        len=2*len;
        change[4]++;
    }
    printf("一共进行了%d 比较: ",compare[4]);
    printf("一共进行了%d 交换: ",change[4]);
} //mergesort end

```

//////////归并函数结束//////////

```

void main()
{

```

```

int i,j,s[N];
printf("1、选择法排序\n");
printf("2、冒泡法排序\n");
printf("3、直接插入法排序\n");
printf("4、快速排序\n");
printf("5、两路合并法排序\n");
printf("7、退出排序比较程序\n");
printf("随机产生的数据:\n");
for(i=1;i<N;i++)
{
    s[i]=rand()%100;
    printf("%6d\n",s[i]);
}
do{
    printf("选择菜单: ");
    printf("\n");
    printf("1、选择法 ,2、冒泡法 ,3、插入法 ,4、快速法 ,5、两路合并法 ,6、退出\n");
    scanf("%d",&j);
    switch(j){
    case 1:
        printf("用选择法排序结果为:");
        simplessort(s);
        break;
    case 2:
        printf("用冒泡法排序结果为:    ");
        bubblesort(s);
        break;
    case 3:
        printf("用直接插入法排序结果为: ");
        insretsort(s);
        break;
    case 4:
        printf("用快速法排序结果为: ");
        q_sort(s,1,N-1);
        break;
    case 5:
        printf("用两路合并法排序结果为: ");
        mergesort(s);break;
    case 6:
        exit(0);break;
    }
} while(1);
}

```