

希尔排序

希尔排序是插入排序的特例。

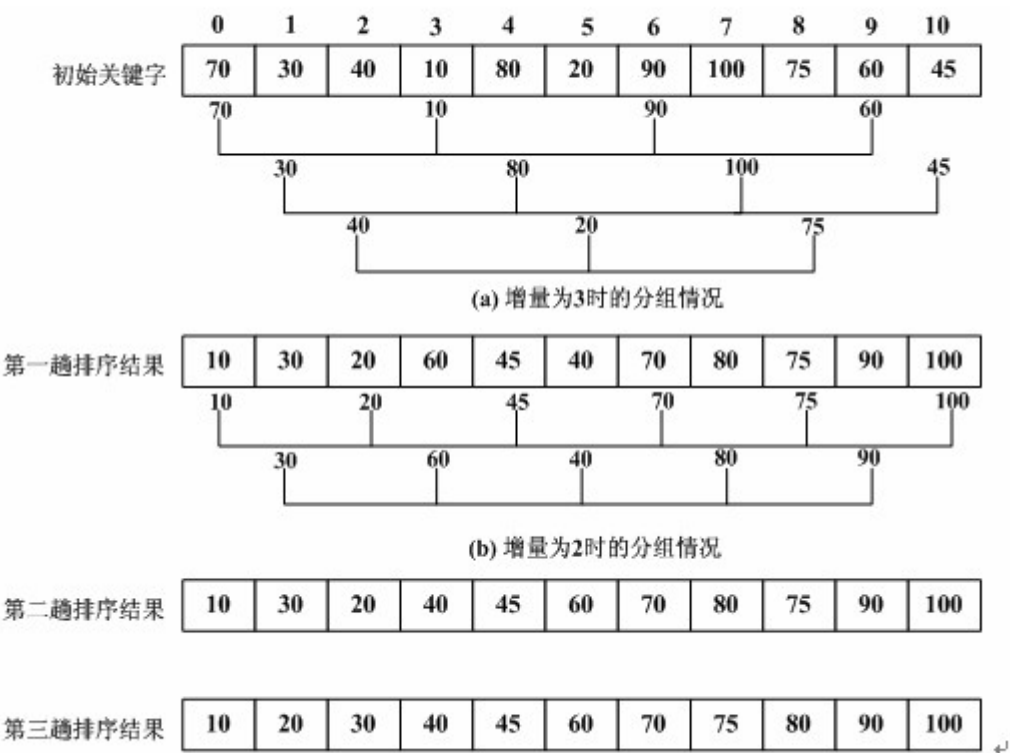
排序规律是，每次选择每个分组中的第一个最小，第二个最小，第三个最小。。。。

希尔排序是一种按照增量排序的方法。其中增量值是小于n的正整数。

shell排序的基本思想[1]是：

先取一个小于n的整数d1作为第一个增量，把文件的全部记录分成d1个组。所有距离为d1的倍数的记录放在同一个组中。先在各组内进行直接插入排序；然后，取第二个增量d2<d1重复上述的分组和排序，直至所取的增量dt=1(dt<dt-1<...<d2<d1)，即所有记录放在同一组中进行直接插入排序为止。

可以根据百度百科中提供的图来直观的看一下：



- (1) 初始增量为3，该数组分为三组分别进行排序。（ 初始增量值原则上可以任意设置（ 0<gap<n ），没有限制）
- (2) 将增量改为2，该数组分为2组分别进行排序。
- (3) 将增量改为1，该数组整体进行排序。

下面是根据该数组实现的代码：

```
#include <iostream>
```

```

using namespace std;

int a[] = {70,30,40,10,80,20,90,100,75,60,45};

void shell_sort(int a[],int n);
int main()
{
    cout<<"Before Sort: ";
    for(int i=0; i<11; i++)
        cout<<a[i]<<" ";
    cout<<endl;
    shell_sort(a, 11);
    cout<<"After Sort: ";
    for(int i=0; i<11; i++)
        cout<<a[i]<<" ";
    cout<<endl;
    system("pause");
}

void shell_sort(int a[], int n)
{
    int gap;
    for(gap = 3; gap >0; gap--)           /// 增量为3，增量为2，增量为1 的分组
    {
        for(int i=0; i<gap; i++)         /// 增量为3时的分组
        {
            for(int j = i+gap; j<n; j=j+gap) /// first group is sorted.
            {
                if(a[j]<a[j-gap])
                {
                    int temp = a[j];      /// tmp is the current value
                    int k = j-gap;        /// previous val of tmp in group
                    while(k>=0&&a[k]>temp)  /// if previous is bigger, this temp a[j] will
move ahead. This group is sorted
                    {
                        a[k+gap] = a[k];
                        k = k-gap;
                    }
                    a[k+gap] = temp;      /// lastly, the temp is find its position, the
previous < temp < a[j]
                }
            }
        }
    }
}

>>>
Before Sort: 70 30 40 10 80 20 90 100 75 60 45
After Sort: 10 20 30 40 45 60 70 75 80 90 100

```

