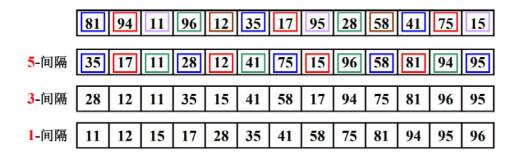
希尔排序

2019年6月29日 19:04

```
增量为n/2
```

```
void ShellSort(int *a,int n)
{
    for (int d=n/2;d>0;d/=2)//增量序列
    {
        int i;
        for (int j = d; j < n; j++)
        {
            int key;//抽的卡
            key = a[j];
            for (i = j; i >=d && a[i - d] > key; i-=d)
            {
                 a[i] = a[i - d];//留空位
            }
            a[i] = key;//此时i是最顶头的一位了
        }
    }
}

最坏情况: T = ·(N2)
```



定义增量序列 DM > DM-1 > ... > D1 = 1 ·

对每个 Dk 进行 "Dk-间隔" 排序(k = M, M-1, ... 1) ·

注意: "Dk-间隔" 有序的序列, 在执行 "Dk-1-间隔" 排序后, 仍然是 "Dk间隔" 有序的

```
Hibbard 增量序列 · D k = 2 k - 1 — 相邻元素互质
最坏情况: T = · (N3/2)
猜想: Tavg = O (N5/4)
Sedgewick增量序列
{1, 5, 19, 41, 109, ...} — 9 · 4 i – 9 · 2 i + 1 或 4 i – 3 · 2 i + 1
猜想: Tavg = O (N7/6), Tworst = O (N4/3)
```

```
整体代码:
#include<iostream>
using namespace std;
void ShellSort(int *a,int n)
      for (int d=n/2;d>0;d/=2)//增量序列
            int i;
        for (int j = d; j < n; j++)
              int key;//抽的卡
              key = a[j];
              for (i = j; i >= d \&\& a[i - d] > key; i-=d)
                    a[i] = a[i - d];//留空位
            a[i] = key;//此时i是最顶头的一位了
      }
}
int main()
      int n;
      cin >> n;
      int* a = new int[n];
      for (int i = 0; i < n; i++)
      {
            cin >> a[i];
      }
      ShellSort(a,n);
      //BubbleSort(a, n);
      //InsertSort(a, n);
     for (int i = 0; i < n; i++)
      {
            cout << a[i];
      }
      delete[] a;
}
```