

# 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:  $T = O(N^2)$

	81	94	11	96	12	35	17	95	28	58	41	75	15
5-间隔	35	17	11	28	12	41	75	15	96	58	81	94	95
3-间隔	28	12	11	35	15	41	58	17	94	75	81	96	95
1-间隔	11	12	15	17	28	35	41	58	75	81	94	95	96

定义增量序列  $D_M > D_{M-1} > \dots > D_1 = 1$

对每个  $D_k$  进行“ $D_k$ -间隔”排序 ( $k = M, M-1, \dots, 1$ )

注意: “ $D_k$ -间隔”有序的序列, 在执行“ $D_{k-1}$ -间隔”排序后, 仍然是“ $D_k$ 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:  $T = O(N^3/2)$
- 猜想:  $T_{avg} = O(N^{5/4})$
- Sedgewick增量序列
- $\{1, 5, 19, 41, 109, \dots\} \rightarrow 9 \cdot 4^{i-9} \cdot 2^{i+1}$  或  $4^{i-3} \cdot 2^{i+1}$
- 猜想:  $T_{avg} = O(N^{7/6})$ ,  $T_{worst} = O(N^{4/3})$

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;
}
```