

插入排序

大连理工大学

于 红

插入排序

- 直接插入排序
- 折半插入排序
- 希尔排序

折半插入排序思想

- 在插入第 i 个记录时，前面的记录已经有序
- 可以用二分法查找第 i 个记录的正确位置

插入排序

折半插入排序演示

例，序列 49 38 65 97 76 13 27

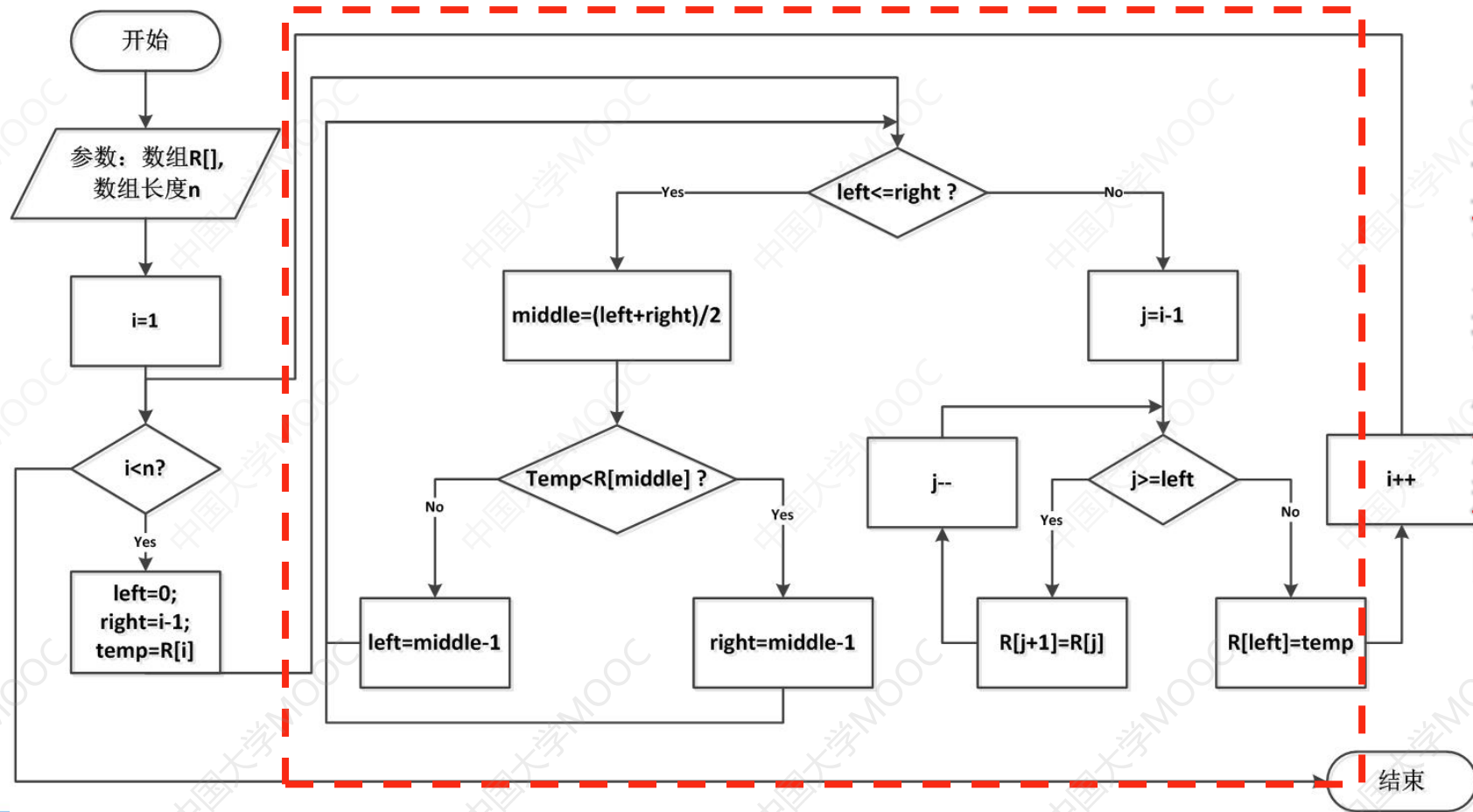
设已形成有序表 { 38 49 65 76 97 }

插入元素 13

↑ right ↑ left ↑ right ↑ mid ↑ right
↑
mid

{ 13 38 49 65 76 97 }

插入排序



折半插入排序流程图

折半插入效率分析

- 时间复杂度仍为 $O(n^2)$ 。
- 稳定的排序。

Shell排序

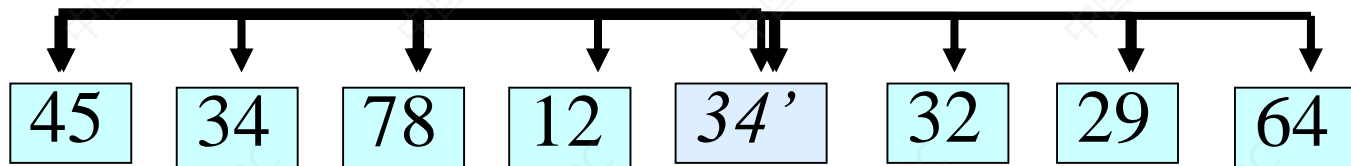
- 直接插入排序的**两个性质**:
 - 在最好情况（序列本身已是有序的）时间代价为 $O(n)$ 。
 - 对于短序列，直接插入排序比较有效。
- Shell排序有效地利用了直接插入排序的这两个性质。

Shell排序思想

- 先将序列转化为**若干小序列**，在这些小序列内进行插入排序。
- 逐渐**扩大小序列**的规模，而减少小序列个数，使得待排序序列逐渐处于更有序的状态。
- 最后对**整个序列**进行扫尾直接插入排序，从而完成排序。

插入排序

Shell排序演示



Shell排序算法

```
template <class T >
void ShellSort (T Vector[], int arrSize ) {
    T temp;
    int gap = arrSize / 2; //gap是子序列间隔
    while ( gap != 0 ) { //循环,直到gap为零
        for ( int i = gap; i < arrSize; i++) {
            temp = Vector[i]; //直接插入排序
            for ( int j = i; j >= gap; j -= gap )
                if ( temp < Vector[j-gap] )
                    Vector[j] = Vector[j-gap];
                else break;
            Vector[j] = temp;
        }
        gap = ( int ) ( gap / 2 );
    }
}
```

Shell排序分析

- 不稳定
- 空间代价： $O(1)$
- 时间代价： 希尔排序的时间复杂性在 $O(n\log_2 n)$ 和 $O(n^2)$ 之间，大致为 $O(n^{1.3})$ 。

插入排序

大连理工大学

于 红