

# 归并排序

大连理工大学

于 红

# 归并排序

## 算法思想

- 简单地将原始序列划分为两个子序列
- 分别对每个子序列递归排序
- 最后将排好序的子序列合并为一个有序序列

# 归并排序

## 算法演示



先划分



再归并

(25 34 45 32 78 12 34' 64)

(25 34 45 32) (78 12 34' 64)

(25 34) (45 32) (78 12) (34' 64)

(25)(34) (45)(32) (78)(12) (34')(64)

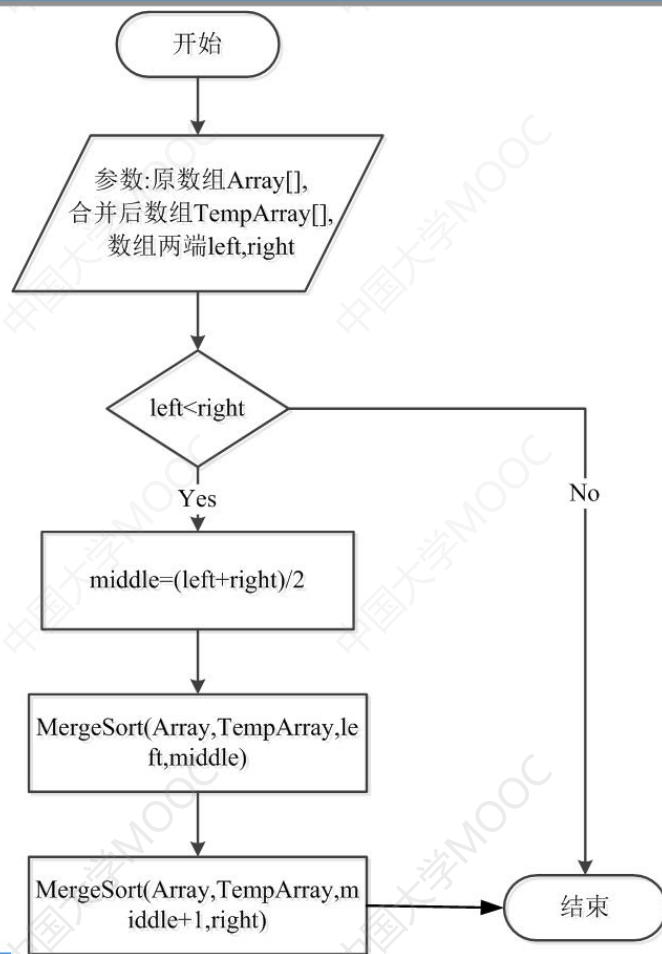
(25 34) (32 45) (12 78) (34' 64)

(25 32 34 45) (12 34' 64 78)

(12 25 32 34 34' 45 64 78)

# 归并排序

## 算法流程图



# 归并排序

## 算法代码

```
template <class Record>
void MergeSort(Record Array[], Record TempArray[],
int left, int right) { // Array为数组, left, right为两端
    int middle;
    if (left < right) { // 序列中只有0或1个记录, 不用排序
        middle = (left + right)/2; // 从中间划为两个子序列
        // 对左边一半进行递归
        MergeSort(Array, TempArray, left, middle);
        // 对右边一半进行递归
        MergeSort(Array, TempArray, middle+1, right);
        Merge(Array, TempArray, left, right, middle); // 归并
    }
}
```

# 归并排序

## 算法分析

- 最坏情况：归并排序是一个递归算法，所以很容易得到算法计算量的递推公式

$$W(n) = W(\lfloor n/2 \rfloor) + W(\lceil n/2 \rceil) + n$$

$$W(1) = 0$$

- 算法最坏情况的复杂度为  $\theta(n \log n)$
- 算法需要  $\theta(n)$  的辅助空间
- 稳定排序

# 归并排序

大连理工大学

于 红