

Lesson 08

分区

Partition

快速选择

QuickSelect

问题

给定一个数组 n

里面不保证排序

6, 1, 7, 8, 2

找到 升序之后 第 k 个数

1, 2, 6, 7, 8

思维

常量降级、倍数降级、分区

方案

```
6 7 3 8 4 0 9 2 1 5
```

Partition:

1. 选出一个 pivot 点

假设选 第 1 个

2. 使用 pivot 点进行 Partition

```
3 4 0 2 1 5 6 7 8 9
    < ^ >
```

Check:

3. 比较 Pivot 当前的位置 和 k

```
when pivot_position > k
when pivot_position < k
when pivot_position = k
```

Recursion:

4. 继续下去

代码:

Partition 算法

方案：

三个变量 pivot, i, j

```
arr[j] > pivot  
j++
```

```
arr[j] < pivot  
swap(arr, i, j)  
i++  
j++
```

时间复杂度分析

最好：

```
arr = 5 1 2 3 4 6 7 8 9  
k = 5
```

第一次 partition 的中轴点（也就是最左边的点 5）

时间 就是 $O(n)$

```
T(n) = 0(n) + c  
      = 0(n)
```

最差的最好：

```
arr = 5 1 2 3 4 6 7 8 9  
k = 4
```

partition 时, pivot 点 都能平分数据
归到 arr 只剩下一个 的时候, 才知道
e

```
T(n) = 0(n) + T(n/2)  
      = 0(n)
```

最差的最差：

```
arr = 1 2 3 4 5 6 7 8 9  
k = 9
```

升序序列, 并且找最后一个

```
T(n) = 0(n) + T(n - 1)  
      = 0(n^2)
```

快排

QuickSort

思路

如果 partition 能确定 一个数的位置

那么我们两边的数据，可以继续 partition

案例

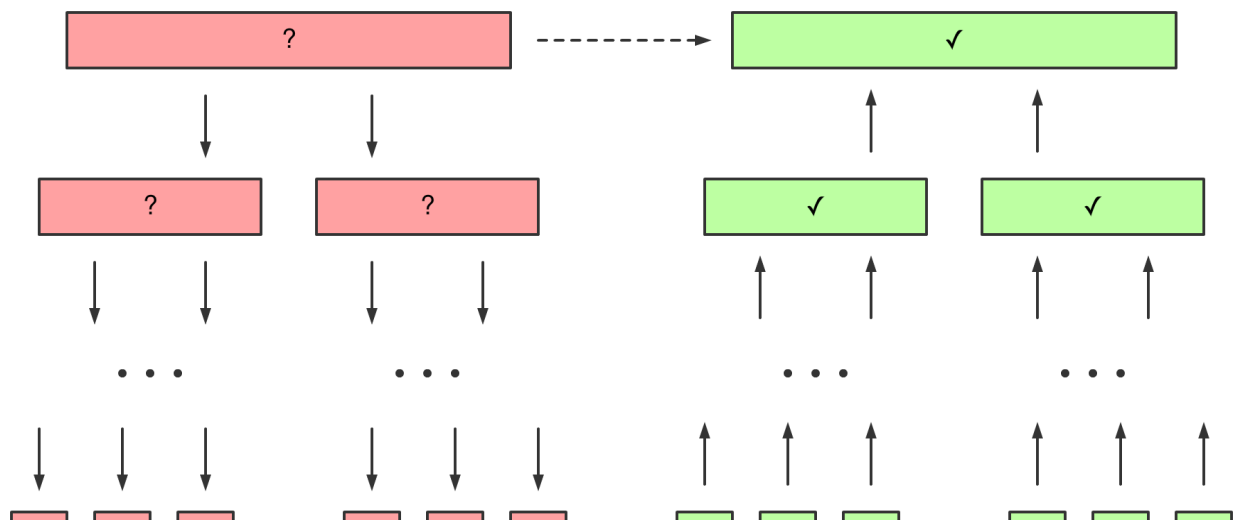
时间复杂度分析

分治

Divide and Conquer

分治思维

思路





不直接解决问题

转而去想

1. 我能不能把问题拆开成几个小部分
2. 然后我能不能把一个小问题解决掉
3. 然后我能不能用小的问题的答案，得出大的问题的答案

核心对应：

1. partition / divide
2. solve $O(1)$
3. merge 合并

重点思考：

当你拿到 两个 一半的答案，能不能得到整体的答案

案例 1

对 n 个数 求 最大值

案例 2 MapReduce

10万个数据的最大值

1台计算机 1天 能计算 1万

需要几天计算完成

归并排序

Merge Sort