快速排序

基本思想

- 用首元素 x 作划分标准，将输入数组 A 划分成不超过 x 的元素构成的数组 A_L，大于 x 的元素构成的数组 A_R. 其中 A_L, A_R 从左到右存放在数组 A 的位置.
- 递归地对子问题 A_L 和 A_R 进行排序，直到子问题规模为 1 时停止.

伪码

算法 Quicksort (A, p, r)

输入：数组$A[p..r]$

输出：排好序的数组 A

1. if $p < r$
2. then $q \leftarrow$Partition (A, p, r)
3. $A[p] \leftrightarrow A[q]$
4. Quicksort (A, p, q-1)
5. Quicksort (A, q+1, r)

初始置 p=1, r=n，然后调用上述算法

划分过程

Partition (A, p, r)

1. $x \leftarrow A[p]$
2. $i \leftarrow p$
3. $j \leftarrow r + 1$
4. **while true do**
5. **repeat** $j \leftarrow j - 1$
6. **until** $A[j] \leq x$ // 不超过首元素的
7. **repeat** $i \leftarrow i + 1$
8. **until** $A[i] > x$ // 比首元素大的
9. **if** $i < j$
10. **then** $A[i] \leftrightarrow A[j]$
11. **else return** j

划分实例

27 99 0 8 13 64 86 16 7 10 88 25 90

i j

27 25 0 8 13 64 86 16 7 10 88 99 90

i j

27 25 0 8 13 10 86 16 7 64 88 99 90

i j

27 25 0 8 13 10 7 16 86 64 88 99 90

j i

<u>16 25 0 8 13 10 7</u> 27 <u>86 64 88 99 90</u>

时间复杂度

最坏情况: $W(n) = W(n-1)+n-1$

$W(1) = 0$

$W(n) = n(n-1)/2$

最好划分: $T(n) = 2\,T(n/2)+n-1$

$T(1) = 0$

$T(n) = \Theta(n\log n)$

均衡划分的时间复杂度

均衡划分：子问题的规模比不变
例如为 1:9

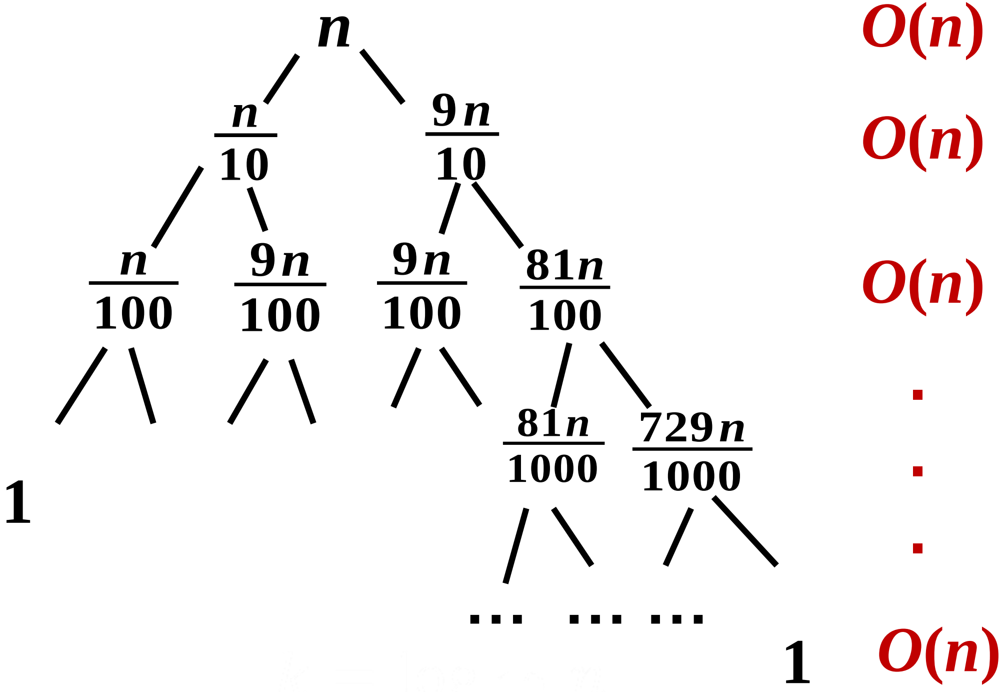
$$T(n) = T(n/10) + T(9n/10) + n$$

$$T(1) = 0$$

根据递归树，时间复杂度

$$T(n) = \Theta(n\log n)$$

递归树



$T(n)=O(n\log n)$

平均时间复杂度

首元素排好序后处在 $1, 2, \ldots, n$

各种情况概率都为 $1/n$

首元素在位置 1： $T(0), T(n-1)$

首元素在位置 2： $T(1), T(n-2)$

....

首元素在位置 n-1： $T(n-2), T(1)$

首元素在位置 n： $T(n-1), T(0)$

子问题工作量 $2[T(1)+T(2)+\ldots+T(n-1)]$

划分工作量 $n-1$

平均时间复杂度

$$T(n)=\frac{1}{n}\sum_{k=1}^{n-1}(T(k)+T(n-k))+n-1$$

$$T(n)=\frac{2}{n}\sum_{k=1}^{n-1}T(k)+n-1$$

$$T(1)=0$$

$$T(n) = \Theta(n\log n)$$

首元素划分后每个位置概率相等

小结

快速排序算法

- 分治策略
- 子问题划分是由首元素决定
- 最坏情况下时间$O(n^2)$
- 平均情况下时间为$O(n\log n)$