

第 9 讲: 排序与选择

姓名: 林凡琪 学号: 211240042

评分: _____ 评阅: _____

April 20, 2022

请独立完成作业, 不得抄袭。
若得到他人帮助, 请致谢。
若参考了其它资料, 请给出引用。
鼓励讨论, 但需独立书写解题过程。

1 作业 (必做部分)

题目 1 (TC 7.2-2)

当数组 A 的所有元素都具有相同值时, QUICKSORT 的时间复杂度是什么?

解答:

此时为最坏情况划分: 0:n-1. 所以此时时间复杂度为 $\Theta(n^2)$

题目 2 (TC 7.3-2)

在 RANDOMIZED-QUICKSORT 的运行过程中, 在最坏情况下, 随机数生成器 RANDOM 被调用了多少次? 在最好情况下呢? 以 Θ 符号的形式给出你的答案.

解答:

因为排列 n 个数最好和最坏情况下都需要选取 $n-1$ 次主元, 所以最好和最坏情况都是 $\Theta(n)$

题目 3 (TC 7.4-2)

证明: 在最好情况下, 快速排序的运动时间为 $\Omega(n \lg n)$

解答:

$$\begin{aligned} T(n) &= 2T\left(\frac{n}{2}\right) + n \\ &= 2\left(2T\left(\frac{n}{4} + \frac{n}{2}\right)\right) + n \\ &= 4T\left(\frac{n}{4}\right) + 2n \\ &= 4T\left(2T\left(\frac{n}{8} + \frac{n}{4}\right)\right) + 2n \\ &= 8T\left(\frac{n}{8}\right) + 3n \end{aligned}$$

因为 $T(1) = 0$, 所以 $T(n) = nT(1) + n * \log n = \Omega(n \lg n)$

题目 4 (TC 8.1-4)

假设现有一个包含 n 个元素的待排序序列

解答:

一个子序列内部有 $k!$ 种可能的排列, 一共有 n/k 个子序列, 所以整个序列一共有 $k!^{n/k}$ 种可能的排列. 设决策树高度为 h , 有 $k!^{n/k} \leq 2^h$
 两边同取对数可得 $h \geq \lg k!^{n/k} = \frac{n}{k} \lg k! = \Omega(n \lg k)$

题目 5 (TC 8.2-4)

设计一个算法, 它能够对于任何给定的介于 0 到 k 之间的 n 个整数先进行预处理, 然后在 $O(1)$ 时间内回答输入的 n 个整数中有多少个落在区间 $[a..b]$ 内. 你设计的算法的预处理时间应为 $\Theta(n + k)$

解答:

假设输入时一个数组 $A[1..n]$, $A.length = n$.

$B[1..n]$ 存放最终排序的输出

$C[1..n]$ 提供临时存储空间, 用于统计数组 A 中每个元素的个数和小于 x 的元素个数

Algorithm 1 SORTING

```

1: function SORTING( $A, B, k$ )
2:   for  $i \leftarrow 0, k$  do
3:      $C[i] = 0$ 
4:   end for
5:   for  $i \leftarrow 1, A.length$  do
6:      $C[A[i]]++$ 
7:   end for
8:   for  $i \leftarrow 1, k$  do
9:      $C[i] += C[i-1]$ 
10:  end for
11:  return  $C[b] - C[a-1]$ 
12: end function

```

题目 6 (TC 8.3-4)

解答:

用基数排序.

把在 $[0..n^3 - 1]$ 区间内的数转化成 n 进制数, 则变成 3 位数. 用基数排序进行排列. 这里用了 3 次计数排序, 每个耗费 $\Theta(n)$, 所以时间复杂度为 $\Theta(n)$

题目 7 (TC Problem 8-2)

Suppose that we have an array of mn data records to sort and that the key of each record has the value 00 or 11. An algorithm for sorting such a set of records might possess some subset of the following three desirable characteristics:

1. The algorithm runs in $O(n)$ time.
2. The algorithm is stable.
3. The algorithm sorts in place, using no more than a constant amount of storage space in addition to the original array.
- a. Give an algorithm that satisfies criteria 1 and 2 above.
 - b. Give an algorithm that satisfies criteria 1 and 3 above.
 - c. Give an algorithm that satisfies criteria 2 and 3 above.
 - d. Can you use any of your sorting algorithms from parts (a)–(c) as the sorting method used in line 2 of RADIX-SORT? RADIX-SORT, so that RADIX-SORT sorts nn records with b -bit keys in $O(bn)$ time? Explain how or why not.
 - e. Suppose that the n records have keys in the range from 1 to k . Show how to modify counting sort so that it sorts the records in place in $O(n+k)$ time. You may use $O(k)$ storage outside the input array. Is your algorithm stable? (*Hint*: How would you do it for $k=3$?)

解答:

- (a) 计数排序
- (b) 快速排序
- (c) 插入排序
- (d) 用 (a) 的算法就可以, 每一次耗费 $O(n)$, 一共有 b 位, 所以时间复杂度 $O(bn) = O(n)$
- (e)

Algorithm 2 SORT

```

1: function MDODIFIED-COUNTING-SORT( $A, k$ )
2:   for  $i \leftarrow 0, k$  do
3:      $C[i] = 0$ 
4:   end for
5:   for  $i \leftarrow 1, A.length$  do
6:      $C[A[i]]++$ 
7:   end for
8:   insert sentinel element NIL at the start of A
9:    $B = C[0..k-1]$ 
10:  insert number 1 at the start of B
11:  for  $i \leftarrow 2, A.length$  do
12:    while  $C[A[i]] \neq B[A[i]]$  do
13:       $key = A[i]$ 
14:       $swap(A[C[A[i]]], A[i])$ 
15:      while  $A[C[key]] == key$  do
16:         $C[key] = C[key] - 1$ 
17:      end while
18:    end while
19:  end for
20:  remove the sentine element
21:  return A
22: end function

```

不稳定

题目 8 (TC 9.1-1)

Show that the second smallest of n elements can be found with $n + \lceil \lg n \rceil - 2n + \lg n - 2$

comparisons in the worst case. (*Hint*: Hint: Also find the smallest element.)

解答:

将元素分成两个大小相等的数组. 每一对两两比较, 只考虑较小的元素. 答案时和最小元素比较的元素或者子问题中第二小的元素. 这样做可以得到最小的和第二小的元素.

$$T(n) = T(n/2) + n/2 + 1$$

$$T(n) \leq n + \lceil \lg n \rceil - 2$$

因为 $T(2) = 1$, 所以得到

$$\begin{aligned} T(n) &= T(n/2) + n/2 + 1 \\ &\leq n/2 + \lceil \lg N/2 \rceil - 2 + n/2 + 1 \\ &= n + \lceil \lg n \rceil - 2 \end{aligned}$$

题目 9 (TC 9.3-7)

解答:

在 $O(n)$ 中找到中位数; 创建一个新数组, 每个元素的绝对值是原始值减去中位数; 在 $O(n)$ 中找到第 k 个最小数, 则所需值是与中位数的绝对差值小于或等于新数组中第 k 个最小数的元素。

2 作业 (选做部分)

题目 1 (TC Problem 7-5)

解答:

3 Open Topics

Open Topics 1 (Analysis of Quicksort)

对 *Quicksort* 更详细的分析。

参考资料:

- [The Analysis of Quicksort Programs \(Sedgewick, 1976\) @ problem-solving-class-paperswelove](#)

Open Topics 2 (Sorting Algorithms)

请介绍其它排序算法或者排序算法的新进展 (比如 TimSort, LibrarySort)。

参考资料:

- [Sorting algorithm @ wiki](#)

4 反馈