

第五周作业-solution

LECTURER: 杨启哲

LAST MODIFIED: 2023 年 12 月 23 日

1. (教材习题 3.5) 请编写一个有效的算法来测试一个给定的数组是否是一个堆。该算法的时间复杂性是多少?

解答. 我们以判断最大堆为例:

算法: 测试数组是否是堆

输入: 数组 $A[1, \dots, n]$

输出: 数组 A 是否是堆

```
1: for  $i = 1$  to  $\lfloor \frac{n}{2} \rfloor$  do
2:   if  $A[i] < A[2i]$  or  $A[i] < A[2i + 1]$  then
3:     return False
4:   end if
5: end for
6: return True
```

算法需要遍历所有的非叶子节点, 因此时间复杂性为 $O(n)$. □

2. (教材习题 3.9) 从一个最大堆中找到最小键值可能有多快?

解答. 由于最大堆的最小值一定在叶子节点上, 并且任何一个叶子节点都有可能是最小值, 所以我们必须要遍历所有的叶子节点, 才能确定其中的最小值. 注意到叶子节点个数可能有 $\frac{n}{2}$ 个, 因此时间复杂度为 $O(n)$. □

3. (k -归并) 给出一个 $O(n \lg k)$ 的算法将 k 个排好序的数组合并成一个排好序的数组. 假设总共有 n 个元素。

Hint: 请考虑利用最小堆进行合并。

解答. 算法思路在于我们每次对这 k 个有序数组取出每个数组当前最小的元素, 构建最小堆, 再不断弹出最小元素即可。

- (1) 首先我们将这 k 个数组的第一个元素构建成一个最小堆, 这一步的时间复杂度为 $O(k)$.
- (2) 然后我们弹出最小元素, 将其所在数组的下一个元素加入到最小堆中, 这一步的时间复杂度为 $O(\log k)$.
- (3) 不断重复上述过程, 直到最小堆为空, 这一共要执行 n 轮, 所以总的复杂度为 $O(n \log k)$.

根据上述讨论, 该算法的时间复杂性为 $O(n \log k)$. □