第 11 讲: 堆与堆排序

姓名: 林凡琪 **学号:** 211240042

评分: ______ **评阅:** ______

2022 年 5 月 4 日

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 6.1-2)
证明有 n 个元素的堆高度为 $\lfloor \lg n \rfloor$

解答:
令 $n = 2^m - 1 + k$, 其中 m 很大.
堆由一个高度为 m-1 的完全二叉树组成, 底部有 k 个附加的叶节点.
根的高度是通往这 k 个叶节点之一的最长的简单路径的长度 m.
从定义可得, $m = \lfloor \lg n \rfloor$

题目 2 (TC 6.1-7)

解答:
取序号为 $\lfloor n/2 \rfloor + 1$ 的子节点.

$$\begin{aligned} \text{LEFT}(\lfloor n/2 \rfloor + 1) &= 2(\lfloor n/2 \rfloor + 1) \\ &> 2(n/2 - 1) + 2 \\ &= n - 2 + 2 \\ &= n. \end{aligned}$$

由于左侧子节点的序号大于堆中元素的个数, 所以该节点没有子节点, 所以是叶节点.
其他有更大区号的节点同理也是叶节点.
但是对于序号为 $\lfloor n/2 \rfloor$ 的节点, 它不是叶节点. 在节点数为偶数的情况下它将有一个序号为 n 的左节点, 在节点数为奇数的情况下, 它将有一个索引为 n-1 的左节点和一个索引为 n 的右节点.
这使得 n 大小的堆中的叶子数量为 $\lceil n/2 \rceil$

题目 3 (TC 6.2-5)

Algorithm 1 MAX-HEAPIFY

解答:

procedure MAX(A, i)
  while 1 **do**
    $l = LEET(i)$
    $r = RIGHT(i)$
    if $l \leq A.heap - sizeandA[l] > A[i]$ **then**
      $largest = l$
    else$largest = i$
    end if
    if $r \leq A.heap - sizeandA[r] > A[largest]$ **then**
      $largest = r$
    end if
    if $largest == i$ **then**
      return
    end if
    $swap(A[i], A[largest])$
    $i = largest$
  end while
end procedure

题目 4 (TC 6.2-6)

解答:

考虑由 A 产生的堆，其中 $A[1] = 1$ $A[i] = 2 (2 \leq i \leq n)$

由于 1 是堆中最小的元素，它必须通过堆的每一层进行交换，直到它成为一个叶子节点。

由于堆的高度是 $lfloor \lg n\rfloor$ 因此MAX-HEAPIFY的最坏情况时间是 $\Omega(\lg n)(lgn)$。

题目 5 (TC 6.3-3)

解答:

从 6.1-7 中，我们知道，堆的叶子节点序号如下

$\lfloor n/2 \rfloor + 1, \lfloor n/2 \rfloor + 2, \ldots, n.$

注意，这些元素对应于堆数组的后半部分（如果 nn 是奇数，则加上中间的元素）。因此，任何大小为 nn 的堆中的叶子数量是 $\lceil n/2 \rceil$ $n/2$ 。

让我们通过归纳法来证明。

让 n_h 表示高度为 h 的节点数. 由于 $n_0 = \lceil n/2 \rceil$ 所以上界对 base 成立.

现在假设它对 $h-1$ 成立. 我们已经证明它对 h 成立.

如果 n_{h-1} 是偶数, 则高度为 h 的每个节点正好有两个子节点, 这意味着 $n_h = n_{h-1}/2 = \lfloor n_{h-1}/2 \rfloor$

如果 n_{h-1} 是奇数, 高度为 h 的一个节点有一个孩子, 其余的有两个孩子, 这也意味着

$n_h = \lfloor n_{h-1}/2 \rfloor + 1 = \lceil n_{h-1}/2 \rceil$
因此, 我们有

$$\begin{aligned} n_h &= \left\lceil \frac{n_{h-1}}{2} \right\rceil \\ &\le \left\lceil \frac{1}{2} \cdot \left\lceil \frac{n}{2^{(h-1)+1}} \right\rceil \right\rceil \\ &= \left\lceil \frac{1}{2} \cdot \left\lceil \frac{n}{2^h} \right\rceil \right\rceil \\ &= \left\lceil \frac{n}{2^{h+1}} \right\rceil \end{aligned}$$

题目 6 (TC 6.4-2)

解答:
Initialization: 数组 A[i+1...n] 是空的, 所以不变式成立
Maintenance:
$A[1]$ 是 A[1...i] 中最大的元素，它比 A[i + 1...n]A[i+1...n] 中的元素小。当我们把它放在第 i 个位置的时候, 那么 $A[i..n]$ 含有已被排序的最大的元素.
减少堆的大小并调用 textMAX-HEAPIFY 将 A[1...i-1] 变成一个最大堆。
递减 i 为下一次迭代奠定了不变性。
Termination:
循环后 i=1。这意味着 A[2...n] 被排序，A[1] 是数组中最小的元素

题目 7 (TC 6.4-4)

解答:
花费线性的时间把它转换为 max-heap, 然后用 $n \lg n$ 的时间排序

题目 8 (TC 6.4-5 (*))

解答:
假设这个堆是个满二叉树,$n = 2^k - 1$, 有 2^{k-1} 个叶子节点和 $2^{k-1} - 1$ 个内部节点.
考虑把堆前面的 2^{k-1} 个元素排序. 我们考虑它们在堆中的排列, 把叶子节点涂成红色, 内部节点涂成蓝色, 有色节点是堆的一个子树. 因为有 2^{k-1} 个有色节点, 所以最多只有 2^{k-2} 个是红色的, 这意味着最少有 $2^{k-2} - 1$ 个节点是蓝色的.
虽然红色节点可以直接跳到根部，但蓝色节点在被移除之前需要向上移动。让我们计算一下将蓝色节点移到根部的最少交换次数。
此时有 $2^{k-2} - 1$ 个节点是蓝色的, 并且它们被排列在二叉树中.
如果有 d 个这样的蓝色节点，那么就会有 $i = \lg d$ 层，每个层包含 2^i 个长度为 i 的节点。因此，互换的次数是

$$\sum_{i=0}^{\lg d} i2^i = 2 + (\lg d - 2)2^{\lg d} = \Omega(d \lg d).$$

有这样的递推关系

$$T(n) = T(n/2) + \Omega(n \lg n).$$

利用 master method 我们得到 $T(n) = \Omega(n \lg n)$

题目 9 (TC 6.5-5)

解答:
Initialiazation:
A 是一个堆，只是 A[i] 可能比它的父代大，因为它已经被修改了。A[i] 比它的子代大，否则 guard clause 会失败，不会进入循环（新值比旧值大，旧值比子代大）。
Maintenance:
当我们将 A[i] 与其父本交换时，max-heap 属性得到了满足，只是现在 A[textPARENT(i)] 可能比其父本大。把 i 改成它的父本，就能保持不变性。
Termination:
只要堆被耗尽或者 A[i] 及其父本的 max-heap 属性被保留，循环就会终止。在循环终止时，A 是一个 max-heap。

题目 10 (TC 6.5-9)

解答:

procedure MERGE-SORTED-LISTS(lists[][])
  n←lists.length
  let lowest-from-each be an empty array
  for $i \leftarrow 1 ton$ **do**
    add (lists[i][0], i) to lowest-from-each
    delete lists[i][0]
  end for
  A←MIN-HEAP(lowest-from-each)
  let merged-lists be an empty array
  while $!min - heap.EMPTY()$ **do**
    element-value, index-of-list ← HEAP-EXTRACT-MIN(A)
    add element-value to merged-lists
    if $lists[index - of - list].length > 0$ **then**
      MIN-HEAP-INSERT(A, lists[index-of-list][0], inde-of-list))
      delete lists[index-of-list][0]
    end if
  end whilereturn merged-lists
end procedure

---

2 作业（选做部分）

题目 1 (Heap Equality)
Prove that

$$\forall h \in \mathbb{Z}^+ : \lceil \log(\lfloor \frac{1}{2} h \rfloor + 1) \rceil + 1 = \lceil \log(h + 1) \rceil.$$

解答:

---

3 Open Topics

Open Topics 1 (Binomial Heaps)
介绍 Binomial Heap 数据结构。
参考资料:

- Problem 19-2 of《算法导论》
- Binomial heap @ wiki

Open Topics 2 (Fibonacci Heaps)
介绍 Fibonacci Heap 数据结构。可不介绍其中的平摊分析内容。
参考资料:

- Chapter 19 of《算法导论》
- Fibonacci heap @ wiki

4 反馈