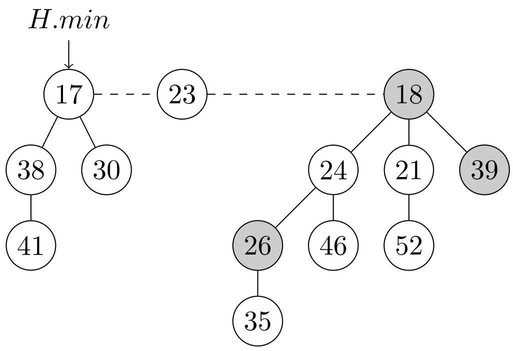
傅申 PB20000051
2022 秋

算法基础 作业 11

19.2–1. 给出图 19–4 (m) 中的斐波那契堆调用 FIB-HEAP-EXTRACT-MIN 后得到的斐波那契堆.

解: 如下



19.3–1. 假定斐波那契堆中的一个根 x 被标记了. 解释 x 是如何成为一个被标记的根的. 试说明 x 是否被标记对分析并没有影响, 即使它不是一个先被链接到另一个结点, 后又丢失了一个孩子的根.

解: x 之前是 $H.min$ 的一个被标记的子结点, 随后执行了 FIB-HEAP-EXTRACT-MIN 操作使得 x 成为了一个被标记的根.

一个结点是否被标记只会在 CASCADING-CUT 的第 3 行中被测试, 但是根结点在其第 2 行测试中就会返回, 因此 x 是否被标记对实际代价并没有影响. 虽然在摊还分析中 x 被标记会使得势函数的值大于实际需要的值, 但是多出来的值并不影响 FIB-HEAP-DECREASE-KEY 的摊还代价.

19.4–2. 假定对级联切断操作进行推广, 对于某个整数常数 k, 只要一个结点失去了它的第 k 个孩子, 就将其从它的父结点上剪切掉 (19.3 节中为 $k = 2$ 的情形). k 取什么值时, 有 $D(n) = O(\lg n)$.

解: 推广后, 引理 19.1 可重写为: 设 x 是堆中的任意结点, 并假定 $x.degree = d$. 设 $y_1, y_2, \ldots, y_d$ 表示以链入先后顺序排序的 x 的孩子结点, 则对 $k-1 \geqslant i \geqslant 1$, 有 $y_i.degree \geqslant 0$; 对 $d \geqslant j \geqslant k$, 有 $y_j.degree \geqslant j-k$.

定义数列 F 为

$$F_m = \begin{cases} 0 & \textbf{if} \quad m = 0 \\ 1 & \textbf{if} \quad k \geqslant m \geqslant 1 \\ F_{m-1} + F_{m-k} & \textbf{if} \quad m \geqslant k+1 \end{cases}.$$

则对于所有的 $m \geqslant 0$, 有

$$F_{m+k} = 1 + \sum_{i=0}^{m} F_i.$$

且若记 α 为方程 $x^k = x^{k-1} + 1$ 的正根, 则对于所有整数 $m \geqslant 0$, 有 $F_{m+k} \geqslant \alpha^m$.

设 s_m 表示堆中度数为 m 的任意结点的最小可能 size. 平凡地, $s_0 = 1$, $s_1 = 2$, …, $s_{k-1} = k$. s_m 随着 m 单调递增. 有 $\text{size}(x) \geqslant s_d$. 考虑 $d \geqslant k$ 的情况, 有

$$\text{size}(x) \geqslant s_d \geqslant k + \sum_{i=k}^{d} s_{y_i.degree} \geqslant k + \sum_{i=k}^{d} s_{i-k}.$$

可以对 d 归纳证明出 $s_d \geqslant F_{d+k} \geqslant \alpha^d$. 因此, 对任意结点 x 都有 $n \geqslant \text{size}(x) \geqslant \alpha^d$, 所以任意结点的最大度数 $D(n)$ 为 $O(\lg n)$. 上面的论证对所有正整数 k 都成立.

21.2–2. 给出下面程序的结果数据结构, 并回答该程序中 FIND-SET 操作返回的答案. 这里使用加权合并启发式策略的链表表示.

```
1  for i = 1 to 16
2      MAKE-SET(x_i)
3  for i = 1 to 15 by 2
4      UNION(x_i, x_{i+1})
5  for i = 1 to 13 by 4
6      UNION(x_i, x_{i+2})
7  UNION(x_1, x_5)
8  UNION(x_11, x_13)
9  UNION(x_1, x_10)
10 FIND-SET(x_2)
11 FIND-SET(x_9)
```

假定如果包含 x_i 和 x_j 的集合有相同的大小, 则 UNION(x_i, x_j) 表示将 x_j 所在的表链接到 x_i 所在的表后.

解: 执行完第 1 行的循环后, 建立了 16 个只含有一个成员 x_i 的集合.

执行完第 3 行的循环后, 共有 8 个集合, 分别为

$$\{x_1 \to x_2\}, \{x_3 \to x_4\}, \{x_5 \to x_6\}, \{x_7 \to x_8\}, \{x_9 \to x_{10}\}, \{x_{11} \to x_{12}\}, \{x_{13} \to x_{14}\}, \{x_{15} \to x_{16}\}.$$

执行完第 5 行的循环后, 共有 4 个集合, 分别为

$$\{x_1 \to x_2 \to x_3 \to x_4\}, \{x_5 \to x_6 \to x_7 \to x_8\}, \{x_9 \to x_{10} \to x_{11} \to x_{12}\}, \{x_{13} \to x_{14} \to x_{15} \to x_{16}\}.$$

第 7 行的 UNION 操作结束后, 共有 3 个集合, 分别为

$$\{x_1 \to x_2 \to x_3 \to x_4 \to x_5 \to x_6 \to x_7 \to x_8\}, \{x_9 \to x_{10} \to x_{11} \to x_{12}\}, \{x_{13} \to x_{14} \to x_{15} \to x_{16}\}.$$

第 8 行的 UNION 操作结束后, 共有 2 个集合, 分别为

$$\{x_1 \to x_2 \to x_3 \to x_4 \to x_5 \to x_6 \to x_7 \to x_8\}, \{x_9 \to x_{10} \to x_{11} \to x_{12} \to x_{13} \to x_{14} \to x_{15} \to x_{16}\}.$$

第 9 行的 UNION 操作结束后, 共有 1 个集合, 为

$$\{x_1 \to x_2 \to x_3 \to x_4 \to x_5 \to x_6 \to x_7 \to x_8 \to x_9 \to x_{10} \to x_{11} \to x_{12} \to x_{13} \to x_{14} \to x_{15} \to x_{16}\}.$$

第 10 行的 FIND-SET(x_2) 和第 11 行的 FIND-SET(x_9) 均返回指向 x_1 的指针.

21.3–3. 给出一个包含 m 个 MAKE-SET, UNION 和 FIND-SET 操作的序列 (其中有 n 个是 MAKE-SET 操作), 当仅使用按秩合并时, 需要 $\Omega(m \lg n)$ 时间.

解: 假定 $m \gg n$. 先执行 n 个 MAKE-SET 操作, 创建 n 个只有一个成员的集合. 然后执行 $2^{\lfloor \lg n \rfloor} - 1$ 次 UNION 操作, 将 $2^{\lfloor \lg n \rfloor}$ 个结点合并为深度为 $\lfloor \lg n \rfloor$ 的树, 其中每次都是选择两棵结点数为 2^k 的树合并为一棵结点数为 2^{k+1} 的树, 这样树中就有且只有一个结点的深度为 $k+1$, 操作的参数均为树根. 最后执行 $m - n - 2^{\lfloor \lg n \rfloor} + 1$ 次 FIND-SET 操作, 每次执行的参数均为深度最高的结点.

MAKE-SET 需要 $\Omega(n)$ 的时间, UNION 需要 $\Omega(n)$ 的时间, FIND-SET 需要 $\Omega((m-2n)\lg n)$ 的时间, 因为 $m \gg n$, 所以总时间为 $\Omega(m \lg n)$.