

## 第 3-4 讲: 用于动态等价关系的数据结构

姓名: 林凡琪 学号: 211240042

评分: \_\_\_\_\_ 评阅: \_\_\_\_\_

2022 年 9 月 28 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 21.1-2)

证明: *CONNECTED - COMPONENTS* 处理完所有的边后, 两个顶点在相同的连通分量中当且仅当它们在同一个集合中。

解答:

必要性: 由 *SAME - COMPONENT* 可知, 若两个元素在同一个 set 里, 那么一定在相同的连通分量中。

充分性: 如果两个顶点  $u$  和  $v$  在同相同的连通分量中, 则  $\exists P(u, v_1), (v_1, v_2), \dots, (v_n, v)$  其中  $v_n \in G.V (n \in N)$ , 又因为在 *CONNECTED - COMPONENTS* 中边的两端都被放进了同一个集合, 所以  $u$  和  $v$  在同一个集合中。

---

#### 题目 2 (TC 21.1-3)

在 *CONNECTED - COMPONENTS* 作用于一个有  $k$  个连通分量的无向图  $G = (V, E)$  的过程中, *FIND - SET* 需要调用多少次? *UNION* 需要调用多少次?

解答:

*FIND - SET* 被调用  $|V| - k$  次 *UNION* 被调用  $k$  次

---

#### 题目 3 (TC 21.2-1)

使用链表表示和加权合并启发式策略, 写出 *MAKE - SET*、*FIND - SET*、和 *UNION* 操作的伪代码。并指定你在集合对象和表对象中所使用的属性。

---

解答:

```

1: procedure MAKE-SET( $X$ )
2:   Create a  $S$ 
3:    $S.head = x$ 
4:    $S.tail = x$ 
5:    $x.next = NIL$ 
6:    $x.p = x$ 
7:    $S.size = 1$ 
8: end procedure
9: procedure UNION( $u, v$ )
10:   $S1 = u.set$ 
11:   $S2 = v.set$ 
12:  if  $S1.size \geq S2.size$  then
13:     $z = S2.head$ 
14:    while  $z \neq NIL$  do
15:       $z.p = S1.head$ 
16:       $z = z.next$ 
17:    end while
18:     $S1.tail = S2.tail$ 
19:    if  $S1.size == S2.size$  then
20:       $S1.size++$ 
21:    end if
22:  else
23:     $UNION\{v, u\}$ 
24:  end if
25: end procedure
26: function FIND-SET( $x$ )
27:  return  $x.set.head$ 
28: end function

```

---

#### 题目 4 (TC 21.2-3)

对定理 21.1 的整体证明进行改造, 得到使用链表表示和加权合并启发式策略下的 MAKE-SET 和 FIND-SET 的摊还时间上界为  $O(\lg n)$

解答:

在定理 21.1 的证明中, 我们得出结论,  $n$  textUNION 操作运行的时间最多为  $O(n \lg n)$ 。这意味着每次最多花费  $O(\lg n)$  的摊销时间。此外, 由于在执行 textMAKE-SET 和 textFIND-SET 操作时只有恒定的实际工作量, 并且这些操作都未用于抵消 textUNION 操作的成本, 因此它们都具有  $O(1)$

---

#### 题目 5 (TC 21.2-6)

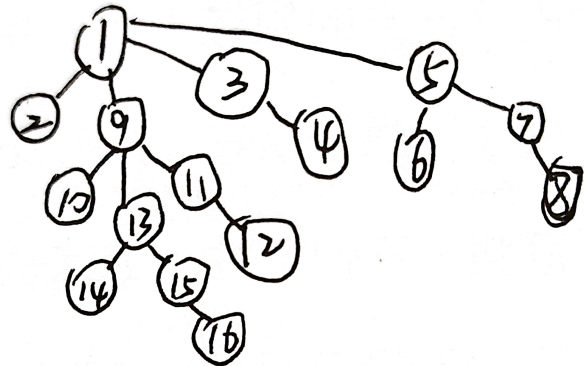
假设对 UNION 过程做一个简单的改动, 在采用链表表示中拿掉让集合对象的 tail 指针总指向每个表的最后一个对象的要求。无论是使用还是不使用加权合并启发式策略, 这个修改不应该改变 UNION 过程的渐近运行时间 (提示: 而不是把一个表链接到另一个表后面, 将它们拼接在一起)

解答:

在 UNION 过程中, 每次和 tail 有关的操作都只占用  $O(1)$ , 其他的操作和 tail 无关, 所以去掉 tail 相关操作并不影响 UNION 过程的渐近运行时间。

**题目 6 (TC 21.3-1)**

用按秩合并与路径压缩启发式策略的不相交集合森林重做练习 21.2-2



解答:

$$FIND-SET(x_2) = x_1 \quad FNID-SET(x_9) = x_1$$

**题目 7 (TC 21.3-2)**

写出使用路径压缩的 FIND-SET 过程的非递归版本。

解答:

```

1: function FIND-SET( $x$ )
2:    $tmp = x$ 
3:   while  $tmp.p \neq tmp$  do
4:      $tmp = tmp.p$ 
5:   end while
6:   while  $x \neq tmp$  do
7:      $x.p = tmp$ 
8:      $x = x.p$ 
9:   end while
10:  return  $tmp$ 
11: end function

```

**题目 8 (TC 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  $\Omega(m \lg n)$  的时间。

---

解答:

```

1: procedure TC 21.3-3
2:   for  $i \leftarrow 1$  to  $n$  do
3:      $MAKE-SET(x[i])$ 
4:   end for
5:   for  $i \leftarrow 1$  to  $k$  do
6:     for  $j \leftarrow 1$  to  $n' - 2^{i-1}by2^i$  do
7:        $UNION(x_i, x_{i+2^{j-1}})$ 
8:     end for
9:   end for
10:  for  $i \leftarrow 1$  to  $m$  do
11:     $FIND-SET(x_1)$ 
12:  end for
13: end procedure

```

---

## 2 作业 (选做部分)

题目 1 (TC Problem 21-1 (Off-line minimum))

解答:

---

## 3 Open Topics

Open Topics 1 (Off-line LCA (TC Problem 21.3))

Open Topics 2 (Partition refinement)

参考资料: [https://en.wikipedia.org/wiki/Partition\\_refinement](https://en.wikipedia.org/wiki/Partition_refinement)

## 4 反馈