

20181109 数据结构作业

1800022769

张靖昆

20181109

我承诺诚实作业，没有抄袭他人！

P159 8.使用重量权衡合并规则与路径压缩，对下列从 0~15 之间的数的等价对进行归并，并给出所得树的父指针表示法的数组表示。在初始情况下，集合中的每个元素分别在独立的等价类中。当两棵树规模同样大时，使结点值较大的根结点作为值较小的根结点的子结点。

(0,2) (1,2) (3,4) (3,1) (3,5) (9,11) (12,14) (3,9) (4,14) (6,7) (8,10) (8,7) (7,0) (10,15)
(10,13)

解：

****记规则**“当两棵树规模同样大时，使结点值较大的根结点作为值较小的根结点的子结点”为规则 1。

****记规则**“重量权衡合并，即根结点不同的两棵树-规模小的树归并到规模大的树上”为规则 2。

****记规则**“路径压缩规则”为规则 3。

****定义单独结点**：单独结点是指没有归入树中独立存在的结点，它们的根结点是它们自身。

****说明 2:** 以{根结点, 结点 1, 结点 2,...}表示一棵树的构成, 仅是为了提升可阅读性。

根据规则 1, 令 θ 为 2 的父节点。

更新结点下标表。

[illegible]

当前已有树: $\{0, 2\}$

根据规则 2，令 1 的父结点为 0。

更新结点下标表。

值(下标同值)	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
---------	---	---	---	---	---	---	---	---	---	---	----	----	----	----	----	----

父结点下标		0	0													
-------	--	---	---	--	--	--	--	--	--	--	--	--	--	--	--	--

当前已有树：{0,2,1}

3. 合并(3,4)：经 find 操作发现 3、4 均是单独结点。

根据规则 1，令 4 的父结点为 3。

更新结点下标表。

值(下标同值)	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
父结点下标		0	0		3											

当前已有树：{0,2,1}、{3,4}

4. 合并(3,1)：经 find 操作发现 3 是单独结点，1 的根结点为 0。

根据规则 2，令 3 的父结点为 0。

更新结点下标表。

值(下标同值)	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
---------	---	---	---	---	---	---	---	---	---	---	----	----	----	----	----	----

父结点下标		0	0	0	3											
-------	--	---	---	---	---	--	--	--	--	--	--	--	--	--	--	--

当前已有树：{0,2,1,3,4}

5. 合并(3,5)：经 find 操作发现 5 是单独结点，3 的根结点为 0。

根据规则 2，令 5 的父结点为 0。

更新结点下标表。

值(下标同值)	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
父结点下标		0	0	0	3	0										

当前已有树：{0,2,1,3,4,5}

6. 合并(9,11)：经 find 操作发现 9、11 均是单独结点。

根据规则 1，令 11 的父结点为 9。

更新结点下标表。

值(下标同值)	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
---------	---	---	---	---	---	---	---	---	---	---	----	----	----	----	----	----

父结点下标		0	0	0	3	0						9				
-------	--	---	---	---	---	---	--	--	--	--	--	---	--	--	--	--

当前已有树：{0,2,1,3,4,5}、{9,11}

7. 合并(12,14)：经 find 操作发现 12、14 均是单独结点。

根据规则 1，令 14 的父结点为 12。

更新结点下标表。

值(下标同值)	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
父结点下标		0	0	0	3	0						9			12	

当前已有树：{0,2,1,3,4,5}、{9,11}、{12,14}

8. 合并(3,9)：经 find 操作发现 9 是一棵规模为 2 的树的根结点，3 的根结点是 0 且树的规模为 6。

根据规则 2，令 9 的父结点为 0。

更新结点下标表。

值(下标同值)	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
---------	---	---	---	---	---	---	---	---	---	---	----	----	----	----	----	----

父结点下标		0	0	0	3	0				0		9			12	
-------	--	---	---	---	---	---	--	--	--	---	--	---	--	--	----	--

当前已有树：{0,2,1,3,4,5,9,11}、{12,14}

9. 合并(4,14)：经 find 操作发现 4 的根结点是 0，所在树规模为 8，14 的根结点是 12，所在树规模为 2。

根据规则 2，令 12 的父结点为 0。

同时，在有关结点 4 的 find 操作中，结点 4 父指针不是根结点 0，根据规则 3，令结点 4 的父节点为 0。

更新结点下标表。

值(下标同值)	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
父结点下标		0	0	0	0	0				0		9	0		12	

当前已有树：{0,2,1,3,4,5,9,11,12,14}

10. 合并(6,7)：经 find 操作发现 6、7 均是单独结点。

根据规则 1，令 7 的父结点为 6。

更新结点下标表。

值(下标同值)	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
父结点下标		0	0	0	0	0		6		0		9	0		12	

当前已有树：{0,2,1,3,4,5,9,11,12,14}、{6,7}

11. 合并(8,10)：经 find 操作发现 8、10 均是单独结点。

根据规则 1，令 10 的父结点为 8。

更新结点下标表。

值(下标同值)	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
父结点下标		0	0	0	0	0		6		0	8	9	0		12	

当前已有树：{0,2,1,3,4,5,9,11,12,14}、{6,7}、{8,10}

12. 合并(8,7)：经 find 操作发现 8 是一颗规模为 2 的树的根结点，7 的根结点是 6，且所在树的规模为 2。

根据规则 1，令 8 的父结点为 6。

更新结点下标表。

值(下标同值)	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
父结点下标		0	0	0	0	0		6	6	0	8	9	0		12	

当前已有树：{0,2,1,3,4,5,9,11,12,14}、{6,7,8,10}

13. 合并(7,0)：经 find 操作发现 0 是一颗规模为 10 的树的根结点，7 的根结点是 6，所在树规模为 4。

根据规则 2，令 6 的父结点为 0。

更新结点下标表。

值(下标同值)	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
父结点下标		0	0	0	0	0	0	6	6	0	8	9	0		12	

当前已有树：{0,2,1,3,4,5,9,11,12,14,6,7,8,10}

14. 合并(10,15)：经 find 操作发现 15 是单独结点，10 的根结点是 0。

根据规则 2，令 15 的父结点下标为 0。

同时，在有关结点 10 的 find 操作中，结点 10、结点 8 的父结点均不是根结点，从而令结点 10、结点 8 的父

结点为 0。

更新结点下标表。

值(下标同值)	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
父结点下标		0	0	0	0	0	0	6	0	0	0	9	0		12	0

当前已有树：{0,2,1,3,4,5,9,11,12,14,6,7,8,10,15}

15. 合并(10,13)：经 find 操作发现 13 是单独结点，10 的根结点是 0。

根据规则 2，令 13 的父结点下标为 0，更新结点下标表。

该表即为重量权衡归并下的父指针表示法的表。

值(下标同值)	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
父结点下标	\	0	0	0	0	0	0	6	0	0	0	9	0	0	12	0

当前已有树：{0,2,1,3,4,5,9,11,12,14,6,7,8,10,15,13}

经过上述归并，得到了经过重量权衡归并得到的归并树，其中 0 为根结点，树结构如下：

