

8.2 用树结构支持并查集的实验范例

吴永辉

Email: yhwu@fudan.edu.cn

WeChat: 13817360465

ICPC Asia Programming Contest 1st Training Committee – Chair

8.2 用树结构支持并查集的实验范例

▶ 并查集

- ▶ 在一些应用中，需要把 n 个不同元素划分成不相交的若干组，每一组的元素构成一个集合，由于这类问题主要涉及对集合的合并和查找，因此称为并查集。
- ▶ 并查集维护一些互不相交的集合 $S=\{S_1, S_2, \dots, S_r\}$ ，每个集合 S_i 都有一个特殊元素 $rep[S_i]$ ，称为集合的代表元。

8.2 用树结构支持并查集的实验范例

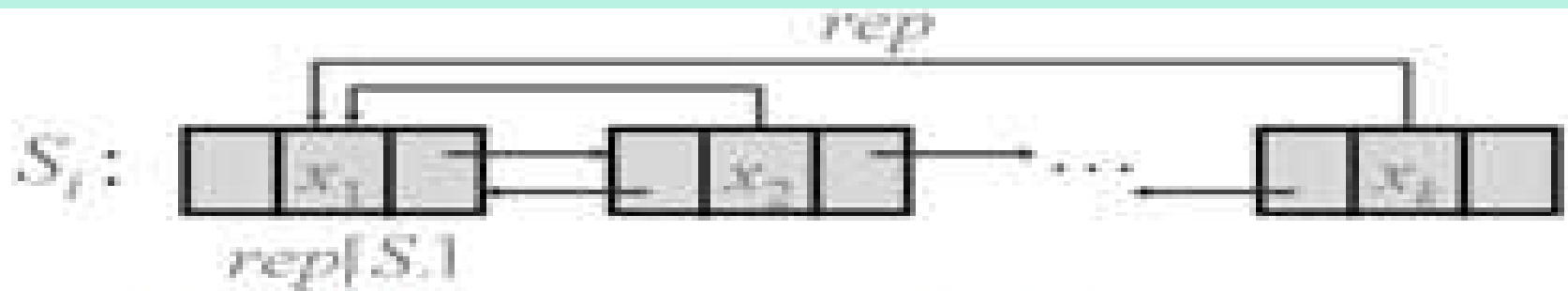
▸ 并查集的三种操作：

- *Make_Set(x)*: 加入一个含单元素 x 的集合 $\{x\}$ 到并查集 S ，且 $rep[\{x\}]=x$ 。
 - x 不能被包含在任何一个 S_i 中，因为 S 里任何两个集合是不相交的。
 - 初始时，对每个元素 x 执行一次*Make_Set(x)*。
- *join(x, y)*: 把 x 和 y 所在的两个不同集合 S_x 和 S_y 合并：从 S 中删除 S_x 和 S_y ，并加入 $S_x \cup S_y$ 。
- *set_find(x)*: 返回 x 所在集合 S_x 的代表元 $rep[S_x]$ 。

8.2 用树结构支持并查集的实验范例

► 链结构

- 每个集合用双向链表表示，代表元 $rep[S_i]$ 在链表首部，集合中的每个节点除前后件指针外，增加一个指向代表元 $rep[S_i]$ 的指针（如图）。

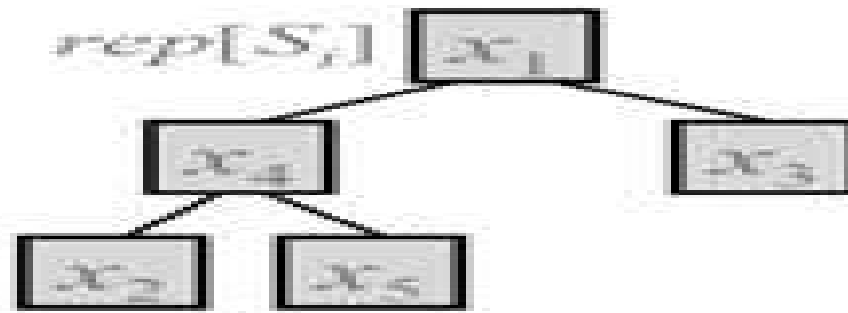


用双向链表存储集合 $S_i = \{x_1, x_2, \dots, x_k\}$

8.2 用树结构支持并查集的实验范例

► 树结构

- 每个集合用一棵树表示，根为集合的代表元。每个节点 p 设一个指针 $set[p]$ ，记录它所在树的根节点序号。如果 $set[p] < 0$ ，则表明 p 为根节点。初始时，为每一个元素建立一个集合，即 $set[x] = -1$ ($1 \leq x \leq n$)。

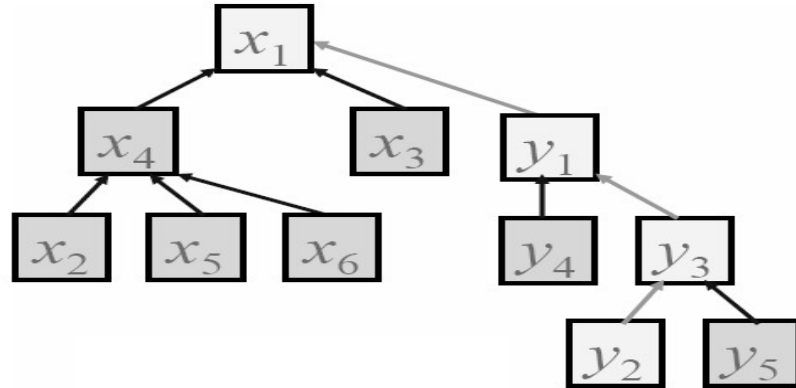


用树存储集合 $S_i = \{x_1, x_2, x_3, x_4, x_5\}$

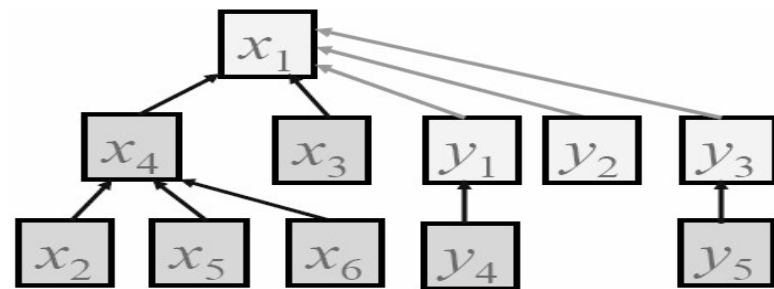
8.2 用树结构支持并查集的实验范例

► 树结构查找操作 $set_find(x)$ ：边查找边“路径压缩”

- 首先，从节点 x 出发，沿 set 指针查找节点 x 所在树的根节点 f （ $set[f] < 0$ ）。
- 然后，进行路径压缩，将 x 至 f 的路径上经过的每个节点的 set 指针都指向 f 。



在原集合中查找 y_2
(a)



查找 y_2 过程中“路径压缩”
(b)

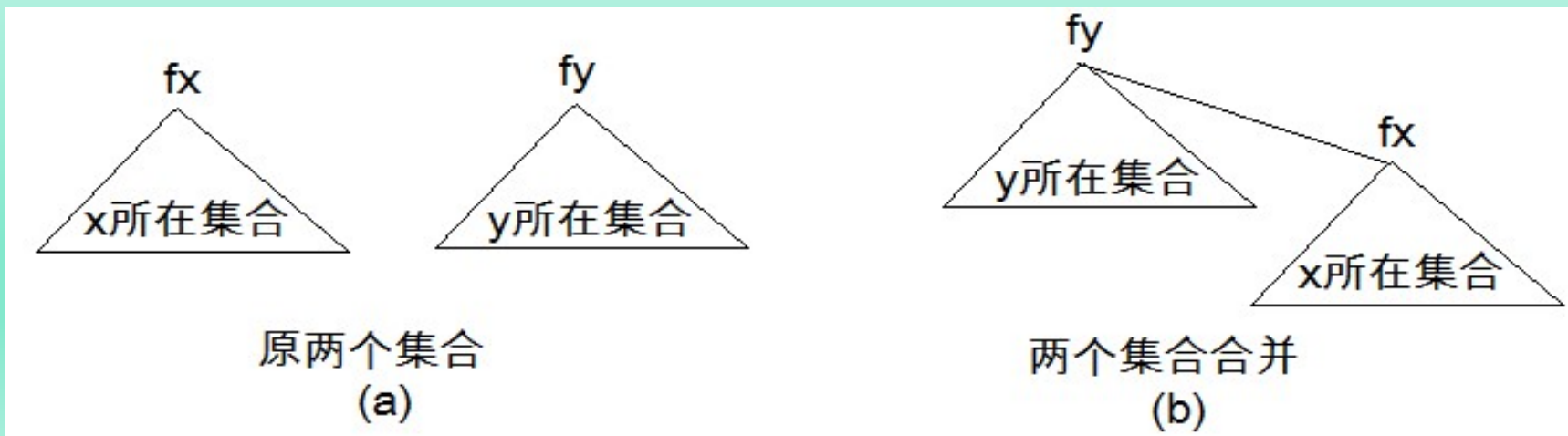
8.2 用树结构支持并查集的实验范例

▶ 树结构查找操作：

```
▶ int set_find(int p)    // 查找p所在集合的代表元，用路径压缩优化
▶ {
▶     if (set[p]<0)
▶         return p;
▶     return set[p]=set_find(set[p]);
▶ }
```

8.2 用树结构支持并查集的实验范例

- ▶ 树结构合并操作 $join(x, y)$ ：将两棵树的根节点相连
 - ▶ 计算 x 元素所在并查集的树根 fx 和 y 元素所在并查集的树根 fy 。如果 $fx == fy$ ，则说明元素 x 和元素 y 在同一并查集中；否则将 x 所在的集合并入 y 所在的集合，也就是将 fx 的 set 指针设为 fy 。



8.2 用树结构支持并查集的实验范例

- ▶ 树结构合并操作：
- ▶ `void join(int p, int q)` // 将 p 所在的集合并入 q 所在的集合
- ▶ {
- ▶ `p=set_find(p);`
- ▶ `q=set_find(q);`
- ▶ if ($p \neq q$)
- ▶ `set[p]=q;`
- ▶ }

8.2.1 Find them, Catch them

- ▶ 试题来源： **POJ Monthly--2004.07.18**
- ▶ 在线测试地址： **POJ 1703**

8.2.1 Find them, Catch them

- ▶ **Tadu**市的警察局决定结束混乱，因此要采取行动，根除城市中的两大帮派：龙帮和蛇帮。然而，警方首先需要确定某个罪犯是属于哪个帮派。目前的问题是，给出两个罪犯，他们是属于同一个帮派吗？您要基于不完全的信息给出您的判断，因为歹徒总是在暗中行事。
- ▶ 假设在**Tadu**市现在有 N （ $N \leq 10^5$ ）个罪犯，编号从1到 N 。当然，至少有一个罪犯属于龙帮，也至少有一个罪犯属于蛇帮。给出 M （ $M \leq 10^5$ ）条消息组成的序列，消息有下列两种形式：
 - ▶ **1. D [a] [b]**
 - ▶ 其中[a]和[b]是两个犯罪分子的编号，他们属于不同的帮派；
 - ▶ **2. A [a] [b]**
 - ▶ 其中[a]和[b]是两个犯罪分子的编号，您要确定a和b是否属于同一帮派。

8.2.1 Find them, Catch them

- ▶ 输入
- ▶ 输入的第一行给出给出一个整数 T （ $1 \leq T \leq 20$ ），表示测试用例的个数。后面跟着 T 个测试用例，每个测试用例的第一行给出两个整数 N 和 M ，后面的 M 行每行给出一条如上面所描述的消息。

8.2.1 Find them, Catch them

- ▶ 输出
- ▶ 对于在测试用例中的每条“A [a] [b]”消息，您的程序要基于此前给出的信息做出判断。回答是如下之一 “In the same gang.”，
“In different gangs.” 或 “Not sure yet.”。

8.2.1 Find them, Catch them

► 解析

- 龙帮和蛇帮的罪犯各组成一个集合，设 $set[d]$ 为罪犯 d 所属集合的代表元， $set[d+n]$ 为另一集合的代表元， $1 \leq d \leq n$ 。函数 $set_find(i)$ 查找罪犯 i 所属并查集的代表元，同时进行路径压缩， $1 \leq i \leq 2n$ 。

8.2.1 Find them, Catch them

- ▶ 初始时 $set[d]=-1$ ，即每个罪犯自成一个帮派。按照如下方法处理每条消息 s ：
- ▶ 确定 a 和 b 是否属于同一帮派（ $s[0]=='A'$ ）
- ▶ 如果 a 和 b 不属同一帮派（ $set_find(a) \neq set_find(b)$ ），且 a 所属的帮派与 b 的另一帮派也不相同（ $set_find(a) \neq set_find(b+n)$ ），则不能确定 a 和 b 是否属于同一帮派；否则，如果罪犯 a 所属集合的代表元与罪犯 b 所属集合的代表元相同（ $set_find(a) = set_find(b)$ ），则确定 a 和 b 同属一个帮派；否则，可以确定 a 和 b 属于不同的帮派。

- 数据结构编程实验：大学程序设计课程与竞赛训练教材（第2版）

8.2.1 Find them, Catch them

- ▶ 设置 a 和 b 分属两个帮派（ $s[0]=='D'$ ）
- ▶ 若 a 所属的帮派不为 b 的另一帮派（ $set_find(a) \neq set_find(b+n)$ ），则 a 的帮派设为 b 的另一帮派， b 的帮派设为 a 的另一帮派（ $set[set_find(a)] = set_find(b+n); set[set_find(b)] = set_find(a+n)$ ）。

- 数据结构编程实验：大学程序设计课程与竞赛训练教材（第2版）

8.2.2 Cube Stacking

- ▶ 试题来源： **USACO 2004 US Open**
- ▶ 在线测试地址： **POJ 1988**

- 数据结构编程实验：大学程序设计课程与竞赛训练教材（第3版）

- ▶ 农夫John和Betsy在玩一个游戏，有 $N(1 \leq N \leq 30,000)$ 块相同的立方，标记从1到 N 。开始时是 N 个栈，每个栈只有一个立方体。农夫John请Betsy执行 $P(1 \leq P \leq 100,000)$ 个操作，有两类操作：move和count。
- ▶ 在一个move操作中，农夫John请Bessie将包含立方体 X 的栈移到包含立方体 Y 的栈的栈顶。
- ▶ 在一个count操作中，农夫John请Bessie计算包含立方体 X 的栈中在 X 下的立方体个数，并返回值。
- ▶ 请您编写一个程序返回游戏结果。

- 数据结构编程实验：大学程序设计课程与竞赛训练教材（第3版）

- ▶ 输入

- ▶ 第1行：一个整数 P 。

- ▶ 第2.. $P+1$ 行：每行给出一个合法的操作，第2行给出第一个操作，依次类推。每行开始时以 ‘M’ 表示一个move操作，或以 ‘C’ 表示一个count操作。对move操作，这一行还给出两个整数： X 和 Y ；对count操作，这一行给出一个整数： X 。

- ▶ 在输入文件中 N 的值不出现。Move操作不会要求一个栈移到它自己的上面。

- ▶ 输出

- ▶ 按输入文件中的次序输出每一个count操作的结果。

- 数据结构编程实验：大学程序设计课程与竞赛训练教材（第3版）

试题解析

- ▶ 每个栈为一个集合，该集合中的元素为栈中的立方体。初始时， n 个栈各放1个立方体。设 $set[k]$ 为元素 k 所在栈的栈底元素序号，亦为该集合的代表元； $cnt[k]$ 为“栈区间” $[k..set[k]]$ 内的元素个数； $top[k]$ 为元素 k 所在栈的栈顶元素序号。

- 数据结构编程实验：大学程序设计课程与竞赛训练教材（第2版）

- ▶ **count**操作：通过函数`set_find(p)`计算 p 所在栈中在 p 下方的元素个数和栈底元素，采用路径压缩优化。
 - ▶ 注意：如果`set[p]`下方还有元素（`set[set[p]] \geq 0`），说明栈区间`[p..set[p]]`的元素移动前栈内有元素。 p 下方的元素个数应调整为`cnt[p]+=cnt[set[p]]`，栈底元素序号应调整为`set[p]=set_find(set[p])`。
- ▶ **move**操作：通过`set_join(x, y)`过程，将 x 所在的栈移到 y 所在的栈顶上：
 - ▶ 首先计算 x 和 y 所在栈的栈底元素（`x=set_find(x); y=set_find(y)`）；调整 x 所在栈的栈底元素（`set[x]=y`）；重新计算原 y 所在栈的栈顶元素到 y 之间的元素个数（`set_find(top[y])`）；将 y 所在栈的栈顶元素更新为 x 原先所在栈的栈顶元素（`top[y]=top[x]`）；调整原 x 所在栈的栈底元素下方的元素数（`cnt[x]=cnt[top[y]]`）。

- 数据结构编程实验：大学程序设计课程与竞赛训练教材（第3版）

8.2.3 食物链

- ▶ 试题来源：NOI 2001
- ▶ 在线测试：POJ 1182

- 数据结构编程实验：大学程序设计课程与竞赛训练教材（第3版）

- ▶ 动物王国中有三类动物A, B, C，这三类动物的食物链构成了有趣的环形：A吃B，B吃C，C吃A。
- ▶ 现有 N 个动物，以 $1 \sim N$ 编号。每个动物都是A, B, C中的一种，但是我们并不知道它到底是哪一种。
- ▶ 有人用两种说法对这 N 个动物所构成的食物链关系进行描述：
- ▶ 第一种说法是" $1 \ X \ Y$ "，表示 X 和 Y 是同类。
- ▶ 第二种说法是" $2 \ X \ Y$ "，表示 X 吃 Y 。

- 数据结构编程实验：大学程序设计课程与竞赛训练教材（第3版）

- ▶ 此人对 N 个动物，用上述两种说法，一句接一句地说出 K 句话，这 K 句话有的是真的，有的是假的。当一句话满足下列三条之一时，这句话就是假话，否则就是真话。
 - ▶ 1) 当前的话与前面的某些真的话冲突，就是假话；
 - ▶ 2) 当前的话中 X 或 Y 比 N 大，就是假话；
 - ▶ 3) 当前的话表示 X 吃 X ，就是假话。
- ▶ 你的任务是根据给定的 N （ $1 \leq N \leq 50,000$ ）和 K 句话（ $0 \leq K \leq 100,000$ ），输出假话的总数。

- 数据结构编程实验：大学程序设计课程与竞赛训练教材（第3版）

- ▶ 输入

- ▶ 第一行是两个整数 N 和 K ，以一个空格分隔。
- ▶ 以下 K 行每行是三个正整数 D ， X ， Y ，两数之间用一个空格隔开，其中 D 表示说法的种类。
- ▶ 若 $D=1$ ，则表示 X 和 Y 是同类。
- ▶ 若 $D=2$ ，则表示 X 吃 Y 。

- ▶ 输出

- ▶ 只有一个整数，表示假话的数目。

- 数据结构编程实验：大学程序设计课程与竞赛训练教材（第3版）

试题解析

- ▶ 本题需要维护、推导集合内元素的关系，所以本题可以利用带权并查集来求解。
- ▶ 数组 pre 和数组 $rela$ 表示集合的关系，其中， pre 表示并查集的代表元， $rela$ 表示集合内元素的关系，本题给出的三类动物的食物链有三种关系：同类，吃，和被吃，显然这种关系是可以量化的，我们分别用0，1，2表示数组 $rela$ 中元素的关系：
 - ▶ 0表示和父节点是同类关系；
 - ▶ 1表示和父节点是吃的关系（吃父节点）；
 - ▶ 2表示和父节点是被吃的关系（被父节点吃）。

- 数据结构编程实验：大学程序设计课程与竞赛训练教材（第3版）

需要维护和推导的关系论述如下：

首先，路径压缩时的关系维护：已知元素 b 和元素 a 的关系，以及元素 a 和所在集合代表元的关系，需要推导出元素 b 和所在集合代表元的关系，如图 8.2-6 所示。

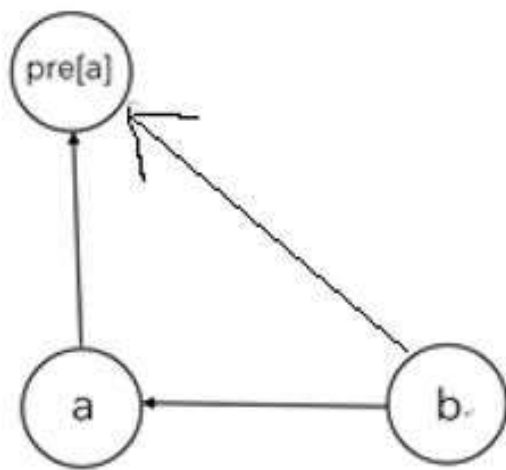


图 8.2-6

- 数据结构编程实验：大学程序设计课程与竞赛训练教材（第3版）

此时，元素 a 和 b 与和所在集合代表元（根）的关系以及元素 b 与元素 a 的关系如下表：

元素 a 与根的关系	元素 b 与元素 a 的关系	元素 b 与根的关系
0	0	0
0	1	1
0	2	2
1	0	1
1	1	2
1	2	0
2	0	2
2	1	0
2	2	1

则关系 $rela[b] = (rela[a] + relation[b \rightarrow a]) \% 3$ 成立。

- 数据结构编程实验：大学程序设计课程与竞赛训练教材（第3版）

然后，元素之间关系的查找：已知元素 a 和元素 b 在同一集合，即它们所在并查集的代表元相同，要求确定元素 a 和元素 b 之间的关系，如图 8.2-7 所示。

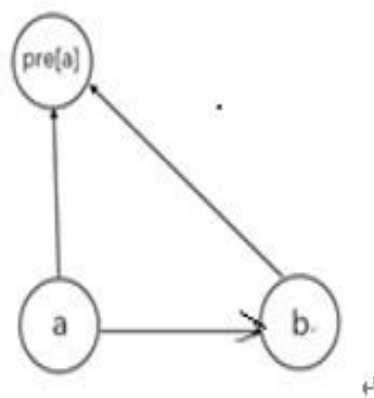


图 8.2-7

- 数据结构编程实验：大学程序设计课程与竞赛训练教材（第3版）

此时，元素 a 和 b 与所在集合代表元（根）的关系以及元素 b 与元素 a 的关系如下表：

元素 a 与根的关系	元素 b 与根的关系	元素 a 与元素 b 的关系
0	0	0
0	1	2
0	2	1
1	0	1
1	1	0
1	2	2
2	0	2
2	1	1
2	2	0

则关系 $relation[a \rightarrow b] = (\text{rel}a[a] - \text{rel}a[b] + 3) \% 3$ 成立。

- 数据结构编程实验：大学程序设计课程与竞赛训练教材（第3版）

最后，两个集合进行并运算时关系的维护。已知元素 a 和其根节点的关系，以及元素 b 和其根节点的关系，以及元素 b 和元素 a 的关系，则当元素 b 和元素 a 所在集合进行并运算时，要给出 b 根节点和 a 根节点存在的关系，关系如图 8.2-8 所示。

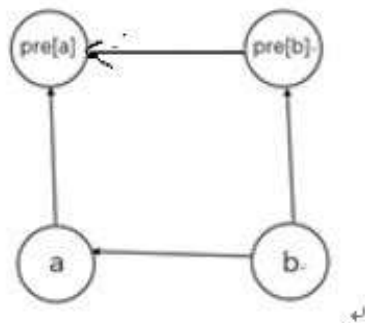
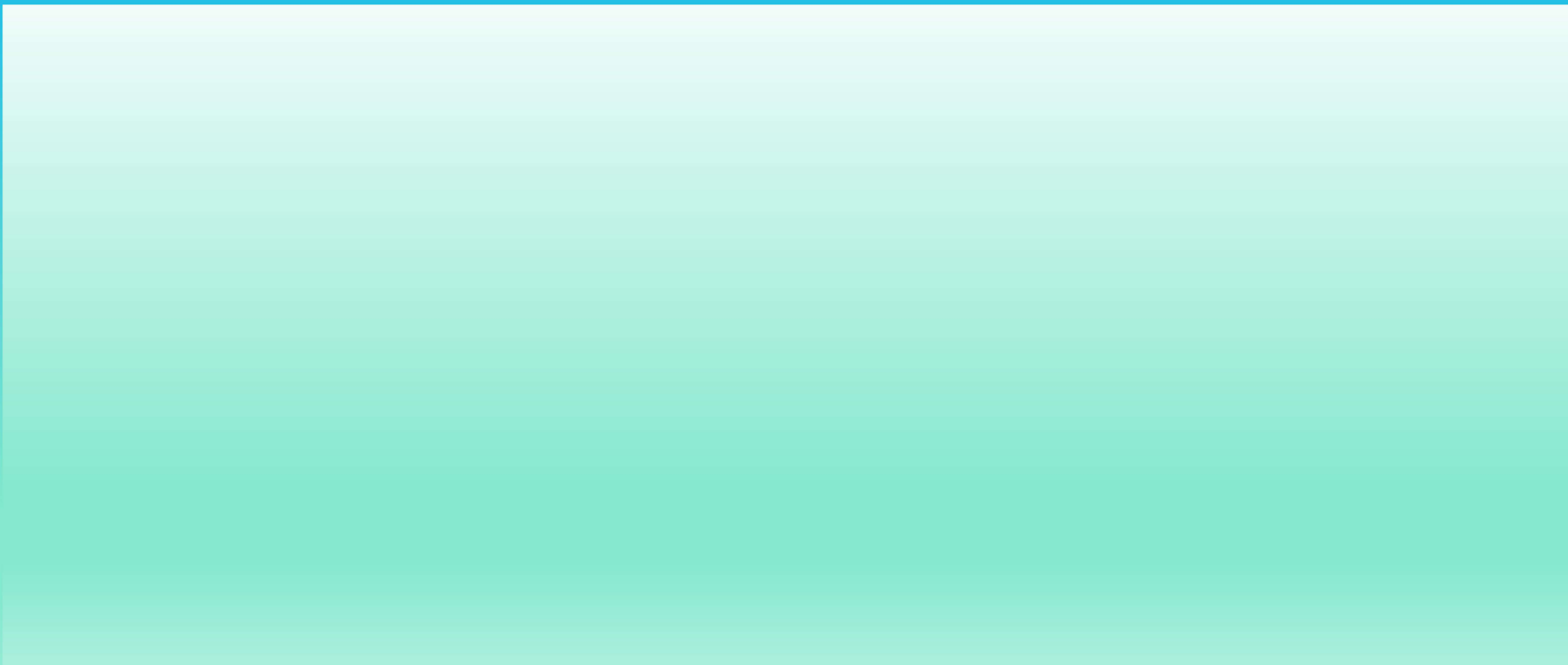


图 8.2-8

$$\text{relation}[pre[b] \rightarrow pre[a]] = (\text{rel}[a] - \text{rel}[b] + \text{relation}[b \rightarrow a]) \% 3$$

- 数据结构编程实验：大学程序设计课程与竞赛训练教材（第3版）



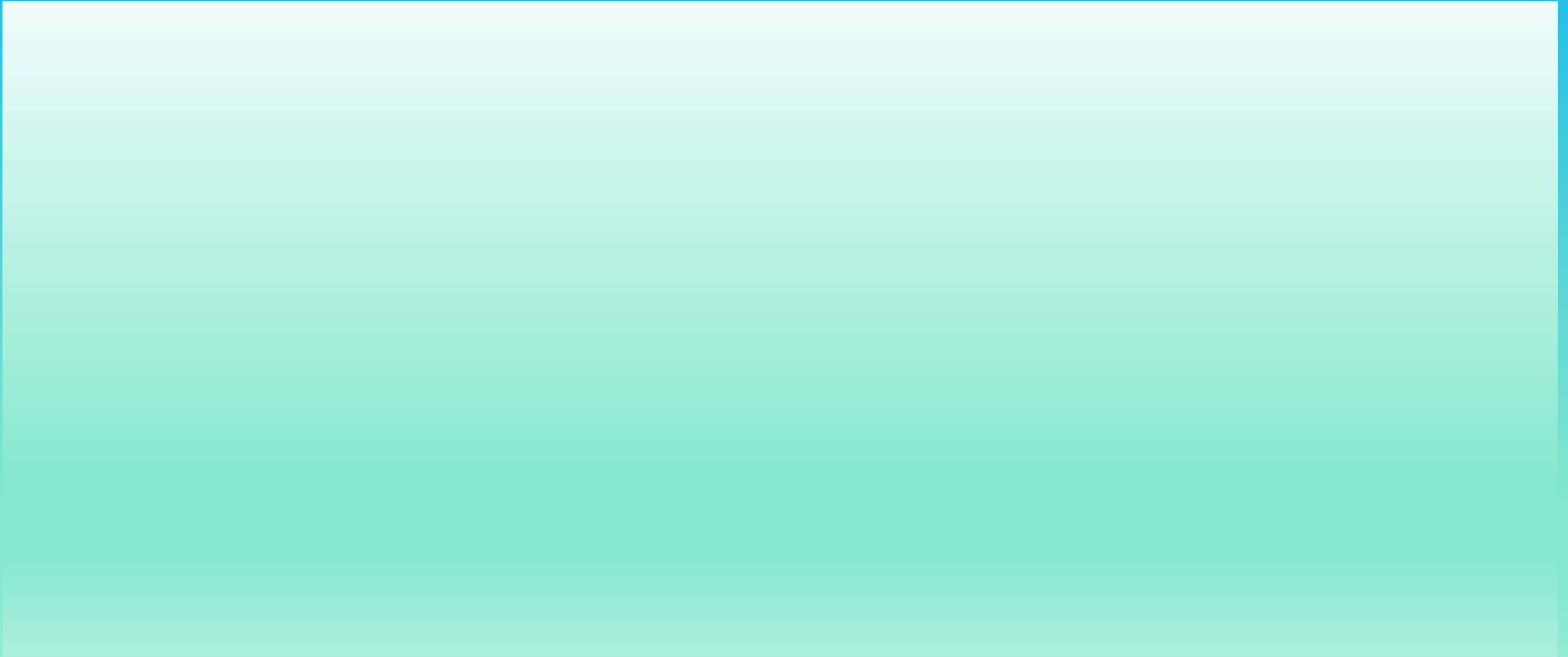
- 数据结构编程实验：大学程序设计课程与竞赛训练教材（第3版）



- 数据结构编程实验：大学程序设计课程与竞赛训练教材（第3版）



- 数据结构编程实验：大学程序设计课程与竞赛训练教材（第3版）



- 数据结构编程实验：大学程序设计课程与竞赛训练教材（第3版）

