



并查集

河南省实验中学  
信息技术组

例题

程序自动分析

超市

打击犯罪

可爱的猴子

走廊泼水节

银河英雄传说

食物链

奇偶游戏

总结

边带权

扩展域

练习

# 并查集题目选讲

河南省实验中学信息技术组

2026年05月15日



# 回顾

## 并查集

河南省实验中学  
信息技术组

- 并查集
- 路径压缩
- 按秩合并

## 例题

程序自动分析

超市

打击犯罪

可爱的猴子

走廊泼水节

银河英雄传说

食物链

奇偶游戏

## 总结

边带权

扩展域

## 练习



## 【例】程序自动分析

并查集

河南省实验中学  
信息技术组

例题

程序自动分析

超市

打击犯罪

可爱的猴子

走廊流水节

银河英雄传说

食物链

奇偶游戏

总结

边带权

扩展域

练习

在实现程序自动分析的过程中，常常需要判定一些约束条件是否能被同时满足。

考虑一个约束满足问题的简化版本：假设  $x_1, x_2, x_3, \dots$  代表程序中出现的变量，给定  $n$  个形如  $x_i = x_j$  或  $x_i \neq x_j$  的变量相等/不等的约束条件，请判定是否可以分别为每一个变量赋予恰当的值，使得上述所有约束条件同时被满足。例如，一个问题中的约束条件为： $x_1 = x_2, x_2 = x_3, x_3 = x_4, x_4 \neq x_1$ ，这些约束条件显然是不可能同时被满足的，因此这个问题应判定为不可被满足。

现在给出一些约束满足问题，请分别对它们进行判定。



## 【例】程序自动分析

并查集

河南省实验中学  
信息技术组

例题

程序自动分析

超市

打击犯罪

可爱的猴子

走廊泼水节

银河英雄传说

食物链

奇偶游戏

总结

边带权

扩展域

练习

### 【输入格式】

输入的第一行包含一个正整数  $T$ ，表示需要判定的问题个数。注意这些问题之间是相互独立的。对于每个问题，包含若干行：

第一行包含一个正整数  $n$ ，表示该问题中需要被满足的约束条件个数。

接下来  $n$  行，每行包括三个整数  $i, j, e$ ，描述一个相等/不等的约束条件，相邻整数之间用单个空格隔开。若  $e = 1$ ，则该约束条件为  $x_i = x_j$ 。若  $e = 0$ ，则该约束条件为  $x_i \neq x_j$ 。

### 【输出格式】

对于每个问题，如果满足条件，输出 YES，否则输出 NO。

### 【数据范围】

对于 70% 的数据， $1 \leq i, j \leq 10^4$ 。

对于 100% 的数据， $1 \leq n \leq 10^5, 1 \leq i, j \leq 10^9, 1 \leq t \leq 10$ 。



## 【例】程序自动分析

并查集

河南省实验中学  
信息技术组

例题

程序自动分析

超市

打击犯罪

可爱的猴子

走廊泼水节

银河英雄传说

食物链

奇偶游戏

总结

边带权

扩展域

练习

### 【样例 1 输入】

```
2
2
1 2 1
1 2 0
2
1 2 1
2 1 1
```

### 【样例 2 输入】

```
2
3
1 2 1
2 3 1
3 1 1
4
1 2 1
2 3 1
3 4 1
1 4 0
```

### 【样例 1 输出】

```
NO
YES
```

### 【样例 2 输出】

```
YES
NO
```



## 【例】程序自动分析

### 并查集

河南省实验中学  
信息技术组

### 例题

程序自动分析

超市

打击犯罪

可爱的猴子

走廊泼水节

银河英雄传说

食物链

奇偶游戏

### 总结

边带权

扩展域

### 练习

- 显然对于相等的约束条件，可以利用并查集并将所有相等的元素加入同一个集合。
- 对于不相等的约束条件，查询是否属于不同的集合即可。
- 由于  $i, j$  范围较大，需要进行离散化。
- 时间复杂度： $O(TN \log N)$ 。



## 【例】程序自动分析

并查集

河南省实验中学  
信息技术组

例题

程序自动分析

超市

打击犯罪

可爱的猴子

走廊泼水节

银河英雄传说

食物链

奇偶游戏

总结

边带权

扩展域

练习

```
1 for(int i = 1; i <= n; ++i) d[i] = a[i], d[i + n] = b[i];
2 sort(d + 1, d + 2 * n + 1);
3 int m = unique(d + 1, d + 2 * n + 1) - (d + 1);
4 for(int i = 1; i <= n; ++i)
5 {
6     a[i] = lower_bound(d + 1, d + m + 1, a[i]) - d;
7     b[i] = lower_bound(d + 1, d + m + 1, b[i]) - d;
8 }
9 for(int i = 1; i <= m; ++i) f[i] = i;
10 for(int i = 1; i <= n; ++i) // 相等 约束条件
11     if(c[i]) merge(a[i], b[i]);
12 bool flag = true;
13 for(int i = 1; i <= n; ++i) // 不相等约束条件
14 {
15     if(c[i]) continue;
16     if(get(a[i]) == get(b[i])) { flag = false; break; }
17 }
```



## 【例】超市

并查集

河南省实验中学  
信息技术组

例题

程序自动分析

超市

打击犯罪

可爱的猴子

走廊泼水节

银河英雄传说

食物链

奇偶游戏

总结

边带权

扩展域

练习

### 【题目描述】

超市里有  $n$  件商品，每个商品都有利润  $p_i$  和过期时间  $d_i$ ，每天只能卖一件商品，过期商品 (在第  $k$  天,  $k > d_i$ ) 不能再卖。

求合理安排每天卖的商品的情况下，可以得到的最大收益是多少。

### 【输入格式】

第 1 行一个整数  $n$  ( $n \leq 10000$ )。

接下来  $n$  行，每行两个整数表示商品的利润和过期时间 ( $1 \leq p_i, d_i \leq 10000$ )。

### 【输出格式】

一行一个整数表示最大收益值。

### 【样例输入】

```
7
20 1
2 1
10 3
100 2
8 2
5 20
50 10
```

### 【样例输出】

```
185
```



## 【例】超市

### 并查集

河南省实验中学  
信息技术组

### 例题

程序自动分析

超市

打击犯罪

可爱的猴子

走廊泼水节

银河英雄传说

食物链

奇偶游戏

### 总结

边带权

扩展域

### 练习

- 在保证不卖出过期商品的前提下，先卖价值高的商品。
- 将商品按照价值从大到小排序，对于商品  $i$ ：
  - 在  $1 \sim d_i$  中的任意一天卖出收益都是一样的，所以安排在第  $d_i$  天卖出；
  - 如果第  $d_i$  天已经有商品卖出了，那么就向前找出能卖商品的一天卖出。
- 时间复杂度： $O(ND)$ 。



## 【例】超市

### 并查集

河南省实验中学  
信息技术组

### 例题

程序自动分析  
超市  
打击犯罪  
可爱的猴子  
走廊泼水节  
银河英雄传说  
食物链  
奇偶游戏

### 总结

边带权  
扩展域

### 练习

- 显然，算法的瓶颈在于需要向前找能卖商品的一天。
- 可以建立一个天数的并查集，起初每一天构成一个并查集。
- 对于商品，它在第  $d$  天过期，那么查询  $d$  的根节点 (记为  $r$ )。
- 若  $r > 0$ ，则把商品安排在第  $r$  天卖出，累加利润，并将  $r$  合并到  $r - 1$  所在的集合。
- 也就是每个并查集的根节点实际上就是离  $d$  最近的没有卖商品的那一天。

```
1 // d 为最大的过期天数
2 sort(a + 1, a + n + 1, cmp);
3 for(int i = 0; i <= d; ++i) f[i] = i;
4 int ans = 0;
5 for(int i = 1; i <= n; ++i)
6 {
7     int r = get(a[i].d);
8     if(r) ans += a[i].p, merge(r - 1, r);
9 }
```

- 时间复杂度:  $O(N \log N)$ 。



## 【例】打击犯罪

并查集

河南省实验中学  
信息技术组

例题

程序自动分析

超市

打击犯罪

可爱的猴子

走廊泼水节

银河英雄传说

食物链

奇偶游戏

总结

边带权

扩展域

练习

### 【题目描述】

某地区有  $n$  个犯罪团伙，当地警方给它们编号  $1 \sim n$ ，它们有些团伙之间有直接或间接联系，这样会形成一个庞大的犯罪集团，犯罪集团的危险程度由犯罪集团内的犯罪团伙的数量唯一确定。

现在警方希望花尽量少的时间（即打击掉尽量少的团伙），使得庞大的犯罪集团分离成若干个较小的集团，并且犯罪集团中最大的一个危险程度不超过  $\frac{n}{2}$ 。

为达到最好效果，警方将按顺序打击掉编号为  $1$  到  $k$  的犯罪团伙，请编程求出  $k$  的最小值。

### 【输入格式】

第一行一个正整数  $n$  ( $n \leq 1000$ )。

接下来  $n$  行，每行由若干个正整数。第  $i$  行第一个整数  $m$  表示犯罪团伙  $i$  和  $m$  个犯罪团伙间由直接联系；接下来  $m$  个整数，表示与  $i$  有直接联系的犯罪团伙。

### 【输出格式】

一个正整数，为  $k$  的值。



## 【例】打击犯罪

并查集

河南省实验中学  
信息技术组

例题

程序自动分析

超市

打击犯罪

可爱的猴子

走廊泼水节

银河英雄传说

食物链

奇偶游戏

总结

边带权

扩展域

练习

### 【样例输入】

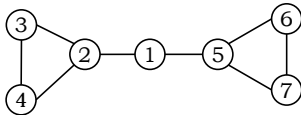
```
7
2 2 5
3 1 3 4
2 2 4
2 2 3
3 1 6 7
2 5 7
2 5 6
```

### 【样例输出】

```
1
```

### 【样例解释】

显然，只需要打击 1 号犯罪团伙即可。





## 【例】打击犯罪

### 并查集

河南省实验中学  
信息技术组

### 例题

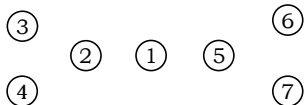
程序自动分析  
超市  
打击犯罪  
可爱的猴子  
走廊泼水节  
银河英雄传说  
食物链  
奇偶游戏

### 总结

边带权  
扩展域

### 练习

- 打击掉犯罪团伙  $i$ ，那么与  $i$  相连的所有边断裂，图可能会分裂为若干个连通子图。
- 并查集只支持**合并**，而不支持分裂，但是分裂的反方向就是合并。
- 首先，令所有犯罪团伙各自单独成为集合；然后从第  $n$  个团伙到第 1 个团伙，逐个增加关系，合并集团，直到某一个集团的大小超过  $n/2$ 。
- 需要统计集团大小，所以需要设置计数数组来记录并查集集合的大小。





## 【例】打击犯罪

并查集

河南省实验中学  
信息技术组

例题

程序自动分析

超市

打击犯罪

可爱的猴子

走廊泼水节

银河英雄传说

食物链

奇偶游戏

总结

边带权

扩展域

练习

```
1 int f[N], d[N]; // d[i] 表示以 i 为根的集合大小
2
3 void merge(int x, int y)
4 {
5     int fx = get(x), fy = get(y);
6     if(fx != fy) f[fy] = fx, d[fx] += d[fy]; // 集合大小合并
7 }
8
9 for(int i = 1; i <= n; ++i) f[i] = i, d[i] = 1;
10 int k = 0;
11 for(int i = n; i >= 1 && !k; --i) // 倒序合并
12 {
13     for(int j = 0; j < g[i].size(); ++j)
14         if(g[i][j] > i) merge(i, g[i][j]); // 编号小的不合并
15     if(d[get(i)] > n / 2) { k = i; break; } // 集团大小超过 n/2
16 }
```



## 【例】可爱的猴子

并查集

河南省实验中学  
信息技术组

例题

程序自动分析

超市

打击犯罪

可爱的猴子

走迷宫水节

银河英雄传说

食物链

奇偶游戏

总结

边带权

扩展域

练习

### 【题目描述】

树上挂着  $n$  只可爱的猴子，编号为  $1, 2, \dots, n$ 。猴子 1 的尾巴挂在树上，每只猴子有两只手，每只手可以最多抓住一只猴子的尾巴。所有的猴子都是悬空的，因此如果一旦脱离了树，猴子会立刻掉到地上。在第  $0, 1, \dots, m - 1$  秒钟某个猴子会把它的某只手松开，因此常常有猴子掉到地上。

现在请你根据这些信息，计算出每个猴子掉在地上的时间。

### 【输入格式】

第一行两个  $n, m$  ( $n \leq 2 \times 10^5, m \leq 4 \times 10^5$ )，表示有  $n$  只猴子，并且总时间为  $m - 1$ 。接下来  $n$  行，描述了每只猴子的信息，每行两个数，分别表示这只猴子左手和右手抓的猴子的编号，如果是  $-1$ ，表示该猴子的那只手没抓其他的猴子。再接下来  $m$  行，按时间顺序给出了一些猴子放手的信息，第  $i$  行表示  $i - 1$  时刻某只猴子的放手信息，信息以两个数给出，前者表示放手的猴子的编号，后者表示其放的是哪只手，1 表示左手，2 表示右手。

### 【输出格式】

共输出  $n$  行，第  $i$  行表示第  $i$  只猴子掉落时刻，若第  $i$  只猴子  $m - 1$  时刻以后还没掉落，就输出  $-1$ 。



## 【例】可爱的猴子

并查集

河南省实验中学  
信息技术组

例题

程序自动分析

超市

打击犯罪

可爱的猴子

走廊泼水节

银河英雄传说

食物链

奇偶游戏

总结

边带权

扩展域

练习

### 【样例输入】

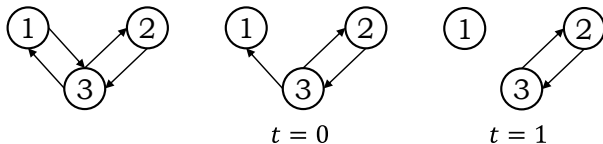
```
3 2
-1 3
3 -1
1 2
1 2
3 1
```

### 【样例输出】

```
-1
1
1
```

### 【样例解释】

- 第 0 秒，1 号猴子松了右手。
- 第 1 秒，3 号猴子送了左手，此时 2 号和 3 号猴子都落地。





## 【例】可爱的猴子

### 并查集

河南省实验中学  
信息技术组

### 例题

程序自动分析

超市

打击犯罪

可爱的猴子

走廊泼水节

银河英雄传说

食物链

奇偶游戏

### 总结

边带权

扩展域

### 练习

- 显然，猴子能否与直接或间接 1 连通是决定其是否落地的条件。
- 随着猴子松手，某些猴子可能开始与 1 不连通，求解的就是这个时刻。
- 如果把连在一起的猴子看成一个集合，每次松手就是断开了集合之间的某些联系或者直接将一个集合分离成两个。



## 【例】可爱的猴子

### 并查集

河南省实验中学  
信息技术组

### 例题

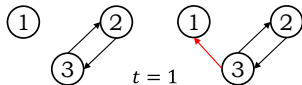
程序自动分析  
超市  
打击犯罪  
可爱的猴子  
走廊流水灯  
银河英雄传说  
食物链  
奇偶游戏

### 总结

边带权  
扩展域

### 练习

- 不妨反过来想，分裂的反向就是合并，如果时间从第  $m - 1$  秒倒流，那么出现的情况是不断有猴子抓住另外一只猴子。
- 于是问题转化为：每只猴子最开始在什么时刻合并到 1 号猴子所在的集合。
- 例如：在  $t = 1$  的时刻 3 抓住 1 合并入 1 所在集合，3 所在的集合全部在  $t = 1$  时掉落。



- 并查集开始时需要将所有不会断开的关系先加入，例如 2, 3 之间的关系。
- 集合在第  $i$  秒合并到 1 所在集合时，集合内所有元素的时间都是  $i$ 。



## 【例】可爱的猴子

### 并查集

河南省实验中学  
信息技术组

### 例题

程序自动分析

超市

打击犯罪

可爱的猴子

走廊泼水节

银河英雄传说

食物链

奇偶游戏

### 总结

边带权

扩展域

### 练习

```
1 int c[N][3]; // 记录猴子之间的连接关系
2 int op[M][2]; // 记录操作
3 bool d[N][3]; // d[i][1] 表示 i 的左手关系是否会断 d[i][2] 表示 i 的右手关系是否会断
4 int ans[N];
5 int f[N]; // 并查集
6
7 for(int i = 0; i < m; ++i) d[op[i][0]][op[i][1]] = true;
8 for(int i = 1; i <= n; ++i) f[i] = 0;
9 for(int i = 1; i <= n; ++i) // 将不会松手的情况先连上
10 {
11     if(!d[i][1] && c[i][1] != -1) merge(i, c[i][1]);
12     if(!d[i][2] && c[i][2] != -1) merge(i, c[i][2]);
13 }
```



## 【例】可爱的猴子

并查集

河南省实验中学  
信息技术组

例题

程序自动分析

超市

打击犯罪

可爱的猴子

走廊泼水节

银河英雄传说

食物链

奇偶游戏

总结

边带权

扩展域

练习

```
1 memset(ans, -1, sizeof(ans));
2 for(int i = m - 1; i >= 0; --i)
3 {
4     int x = op[i][0], y = c[x][op[i][1]];
5     int fx = get(x), fy = get(y);
6     if(fx == fy) continue;
7     if(fx == get(1)) // fy 合并到集合 1
8         for(int k = 1; k <= n; ++k) if(get(k) == fy) ans[k] = i;
9     if(fy == get(1)) // fx 合并到集合 1
10        for(int k = 1; k <= n; ++k) if(get(k) == fx) ans[k] = i;
11    merge(fx, fy);
12 }
```

- 上述算法的时间复杂度为  $O(NM)$ ，会超时。



## 【例】可爱的猴子

### 并查集

河南省实验中学  
信息技术组

### 例题

程序自动分析  
超市  
打击犯罪  
可爱的猴子  
走廊 watering  
银河英雄传说  
食物链  
奇偶游戏

### 总结

边带权  
扩展域

### 练习

- 显然每一次记录时间都需要遍历所有元素来判定元素是否属于集合，这是效率低的原因。
- 那么可以只在集合的根结点上存储时间，其代表的集合的其他元素的时间都直接查询根即可。
- 但是要注意，集合合并到 1 上时，集合所有元素查询的时间都是 1 的时间，所以并查集不能合并到 1 所在集合，故而需要新建一个并查集来处理时间问题。

```
1 int f[N]; // 对于每个操作都要合并
2 int g[N]; // 对于两个集合都不是 1 的情况下的合并，用于记录集合第一次合并到 1 的时间
3
4 for(int i = 0; i < m; ++i) d[op[i][0]][op[i][1]] = true;
5 for(int i = 1; i <= n; ++i) f[i] = i;
6 for(int i = 1; i <= n; ++i) // 将不会松手的情况先连上
7 {
8     if(!d[i][1] && c[i][1] != -1) merge(f, i, c[i][1]);
9     if(!d[i][2] && c[i][2] != -1) merge(f, i, c[i][2]);
10 }
```



## 【例】可爱的猴子

### 并查集

河南省实验中学  
信息技术组

### 例题

程序自动分析  
超市  
打击犯罪  
可爱的猴子  
走廊泼水节  
银河英雄传说  
食物链  
奇偶游戏

### 总结

边带权  
扩展域

### 练习

```
1 for(int i = 1; i <= n; ++i) g[i] = f[i]; // 并查集拷贝一份
2 memset(ans, -1, sizeof(ans));
3 for(int i = m - 1; i >= 0; --i)
4 {
5     int x = op[i][0], y = c[x][op[i][1]];
6     int fx = get(f, x), fy = get(f, y);
7     if(fx == fy) continue;
8     if(fx == get(f, 1)) ans[fy] = i; // 有一个集合是 1 所在集合
9     else if(fy == get(f, 1)) ans[fx] = i;
10    else merge(g, x, y); // 两个集合都不是 1
11    merge(f, fx, fy);
12 }
13 for(int i = 1; i <= n; ++i) printf("%d\n", ans[get(g, i)]);
```

- 上述算法的时间复杂度为  $O(M)$ 。



## 【例】走廊泼水节

并查集

河南省实验中学  
信息技术组

例题

程序自动分析

超市

打击犯罪

可爱的猴子

走廊泼水节

银河英雄传说

食物链

奇偶游戏

总结

边带权

扩展域

练习

### 【题目描述】

有  $n$  个人打算参加泼水节，同时很凑巧的是正好有  $n$  个水龙头。 $n$  个水龙头之间正好有  $n - 1$  条小道，并且每个水龙头都可以经过小道到达其他水龙头。但是人们决定修建一些个道路，使得每个水龙头到每个水龙头之间都有一条直接的道路连接，并且希望所有水龙头之间修建的道路，都要大于原来两个水龙头之间路径上的所有小道的长度。请你计算修建的那些道路总长度最短是多少。

### 【输入格式】

第一行一个整数  $n$  ( $n \leq 6000$ )，表示人和水龙头的个数。

接下来  $n - 1$  行，每行三个整数  $x, y, z$  ( $1 \leq x, y \leq n, 1 \leq z \leq 100$ )，表示水龙头  $x$  和水龙头  $y$  之间有一条长度为  $z$  的小道。

### 【输出格式】

输出一个整数，表示修建的所有道路总长度的最短值。



# 【例】走廊泼水节

并查集

河南省实验中学  
信息技术组

例题

程序自动分析

超市

打击犯罪

可爱的猴子

走廊泼水节

银河英雄传说

食物链

奇偶游戏

总结

边带权

扩展域

练习

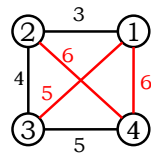
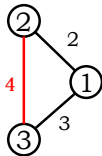
## 【样例 1 输入】

```
3
1 2 2
1 3 3
```

## 【样例 2 输入】

```
4
1 2 3
2 3 4
3 4 5
```

## 【样例解释】



## 【样例 1 输出】

```
4
```

## 【样例 2 输出】

```
17
```



## 【例】走廊泼水节

并查集

河南省实验中学  
信息技术组

例题

程序自动分析

超市

打击犯罪

可爱的猴子

走廊泼水节

银河英雄传说

食物链

奇偶游戏

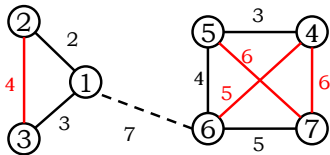
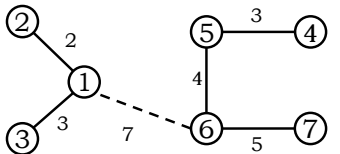
总结

边带权

扩展域

练习

- 新加入的边的权值都比原来的权值大，而且任意两个顶点之间都有边，那么也就是从最小生成树扩充为完全图。
- 为了保证增加的边权总和最小，那么增加的边要比原来最小生成树的路径上的边大 1 即可。
- 在如下的最小生成树中，将 1, 2, 3 和 4, 5, 6, 7 分为两块，可以发现两块一直扩充到完全图之前是互不影响的。





## 【例】走廊泼水节

并查集

河南省实验中学  
信息技术组

例题

程序自动分析

超市

打击犯罪

可爱的猴子

走廊泼水节

银河英雄传说

食物链

奇偶游戏

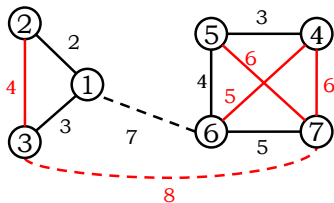
总结

边带权

扩展域

练习

- 当左右两块都成为完全图后，下一次需要连接左右两块之间的顶点，这条增加的边(红色虚线)必然一个顶点在1,2,3中，另一个顶点在4,5,6,7中，权值也一定为8，总共有  $3 \times 4 - 1$  条，故而对答案的贡献为  $(3 \times 4 - 1) \times 8$ 。



- 按照最小生成树的算法，当枚举到边  $(x, y, z)$  进行并查集合并时，增加的边对答案的贡献为  $(d_x \times d_y - 1) \times (z + 1)$ ，其中  $d_x, d_y$  分别为  $x, y$  所在并查集集合大小。



## 【例】走廊泼水节

### 并查集

河南省实验中学  
信息技术组

### 例题

程序自动分析  
超市  
打击犯罪  
可爱的猴子  
走廊泼水节  
银河英雄传说  
食物链  
奇偶游戏

### 总结

边带权  
扩展域

### 练习

```
1 for(int i = 1; i <= n; ++i) f[i] = i, d[i] = 1;
2 sort(e + 1, e + n, cmp);
3 long long ans = 0;
4 for(int i = 1; i <= n - 1; ++i)
5 {
6     int x = get(e[i].x), y = get(e[i].y);
7     long long z = e[i].z;
8     ans += ((long long)d[x] * d[y] - 1) * (z + 1);
9     merge(x, y);
10 }
```



## 【例】银河英雄传说

### 并查集

河南省实验中学  
信息技术组

### 例题

程序自动分析  
超市  
打击犯罪  
可爱的猴子  
走廊泼水节  
银河英雄传说  
食物链  
奇偶游戏

### 总结

边带权  
扩展域

### 练习

有一个划分为 30000 列的星际战场，各列依次编号为  $1, 2, \dots, 30000$ 。同样，有 30000 艘战舰，也依次编号为  $1, 2, \dots, 30000$ ，其中第  $i$  号战舰处于第  $i$  列。

有  $T(1 \leq T \leq 5 \times 10^5)$  条指令，每条指令格式为以下两种之一：

- ①  $M \ i \ j$ ，表示让第  $i$  号战舰所在列的全部战舰保持原有顺序，接在第  $j$  号战舰所在列的尾部。
  - ②  $C \ i \ j$ ，表示询问第  $i$  号战舰与第  $j$  号战舰当前是否处于同一列中，如果不在同一列，输出  $-1$ ，如果在同一列中，输出它们之间间隔了多少艘战舰。
- 现在需要你编写一个程序，处理一系列的指令。



## 【例】银河英雄传说

并查集

河南省实验中学  
信息技术组

例题

程序自动分析

超市

打击犯罪

可爱的猴子

走廊泼水节

银河英雄传说

食物链

奇偶游戏

总结

边带权

扩展域

练习

### 【输入格式】

第一行一个整数  $T$ ，表示指令条数。

接下来  $T$  行，每行一条指令，具体格式见题目描述。

### 【输出格式】

对于询问指令，输入一行一个整数。

如果第  $i$  号战舰和第  $j$  号战舰在同一列，输出它们之间间隔的战舰数；如果不在同一列，输出  $-1$ 。

### 【样例输入】

```
4
M 2 3
C 1 2
M 2 4
C 4 2
```

### 【样例输出】

```
-1
1
```



## 【例】银河英雄传说

并查集

河南省实验中学  
信息技术组

例题

程序自动分析

超市

打击犯罪

可爱的猴子

走廊流水节

银河英雄传说

食物链

奇偶游戏

总结

边带权

扩展域

练习

- 模拟法
- 使用链表来维护每一个战舰列，在链表中每一个战舰记录其前方战舰编号。
- 定义数组  $f$ ，其中  $f[x]$  表示战舰  $x$  前方战舰编号。
- 合并战舰列时，只需要将第  $i$  号战舰所在链表表头指向第  $j$  号战舰所在链表表尾即可。
- 查询时，战舰  $i$  和战舰  $j$  在链表中的位置之差就是二者之间的战舰数量；如果不在同一个链表中，则输出  $-1$  即可。
- 由于数据量较大，所以实现时运行时间过长。



## 【例】银河英雄传说

### 并查集

河南省实验中学  
信息技术组

### 例题

程序自动分析  
超市  
打击犯罪  
可爱的猴子  
走廊 watering  
银河英雄传说  
食物链  
奇偶游戏

### 总结

边带权  
扩展域

### 练习

- 在本题中，我们不关系某一战舰前方具体是哪一个战舰，而重点关心两个战舰是否在同一列以及他们的位置。
- 因此，我们将每一列战舰看作是一个集合，用并查集维护，一个集合的代表就是位于最前方的战舰。
- 另外，建立一个数组  $d$ ，其中  $d[x]$  记录战舰  $x$  与  $f[x]$  之间的距离，那么从战舰  $x$  到根结点路径长度就是战舰  $x$  在列中位置。



## 【例】银河英雄传说

并查集

河南省实验中学  
信息技术组

例题

程序自动分析

超市

打击犯罪

可爱的猴子

走廊泼水节

银河英雄传说

食物链

奇偶游戏

总结

边带权

扩展域

练习

- 在查找战舰  $x$  所属列代表元时，把  $x$  直接指向根结点，同时把  $d[x]$  更新为从  $x$  到根结点的路径长度 ( $x$  的位置)。

```
1 int get(int x)
2 {
3     if(x == f[x]) return x;
4     int fx = get(f[x]);
5     d[x] = d[x] + d[f[x]];
6     return f[x] = fx;
7 }
```

- 当执行  $C \ x \ y$  指令时，分别执行  $get(x)$  和  $get(y)$  完成查询和路径压缩。
- 若二者返回值相同，说明  $x$  和  $y$  在同一列中。
- 此时  $d[x]$  保存了  $x$  之前的战舰数量， $d[y]$  保存了  $y$  之前的战舰数量，二者距离之差减一，即为  $x$  和  $y$  之间间隔的战舰数量。



## 【例】银河英雄传说

### 并查集

河南省实验中学  
信息技术组

### 例题

程序自动分析

超市

打击犯罪

可爱的猴子

走廊泼水节

银河英雄传说

食物链

奇偶游戏

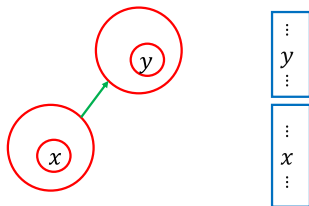
### 总结

边带权

扩展域

### 练习

- 当执行  $M \times y$  指令时, 把  $x$  所在集合合并到  $y$  所在的集合。
- 此时要注意  $x$  所在集合的代表元的  $d$  值应该设置为  $y$  所在集合大小, 因此需要一个数组来记录集合大小。



```
1 int sz[N + 10]; // 集合大小
2 void merge(int x, int y)
3 {
4     int fx = get(x), fy = get(y);
5     f[fx] = fy, d[fx] = sz[fy], sz[fy] += sz[fx];
6 }
```



## 【例】食物链

并查集

河南省实验中学  
信息技术组

例题

程序自动分析

超市

打击犯罪

可爱的猴子

走麻渡水节

银河英雄传说

食物链

奇偶游戏

总结

边带权

扩展域

练习

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  $1 \times y$ , 表示  $x$  和  $y$  是同类。
- 第二种说法是  $2 \times y$ , 表示  $x$  吃  $y$ 。

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  $n$  和  $k$  句话, 输出假话的总数。



## 【例】食物链

并查集

河南省实验中学  
信息技术组

例题

程序自动分析

超市

打击犯罪

可爱的猴子

走廊泼水节

银河英雄传说

食物链

奇偶游戏

总结

边带权

扩展域

练习

### 【输入格式】

第一行两个整数  $n, k (1 \leq n \leq 5 \times 10^4, 0 \leq k \leq 10^5)$ 。

接下来  $k$ ，每行一个描述，具体格式见题目描述。

### 【输出格式】

一行一个整数，表示假话的数目。

### 【样例输入】

```
100 7
1 101 1
2 1 2
2 2 3
2 3 3
1 1 3
2 3 1
1 5 5
```

### 【样例输出】

```
3
```



## 【例】食物链

并查集

河南省实验中学  
信息技术组

例题

程序自动分析

超市

打击犯罪

可爱的猴子

走廊泼水节

银河英雄传说

食物链

奇偶游戏

总结

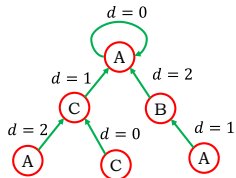
边带权

扩展域

练习

- 方法 1: 边带权并查集
- 建立一个带权并查集, 其中边权为结点与父结点的关系。对于  $x$  和  $y$  两个动物, 其中  $f[x] = y$ , 他们之间的关系有

- $x$  和  $y$  是同类, 则  $d[x] = 0$ ;
- $x$  吃  $y$ , 则  $d[x] = 1$ ;
- $x$  被  $y$  吃, 则  $d[x] = 2$ 。



- 可以进行路径压缩, 在压缩的过程中更新距离。对于  $x$ , 如果距离  $d[x] \% 3 = 0$ , 则它与根结点同类; 如果距离  $d[x] \% 3 = 1$ , 则它吃根结点; 如果距离  $d[x] \% 3 = 2$ , 则它被根结点吃。
- 那么, 对于动物  $x$  和  $y$  (假设它们已经在同一个集合中), 如何判定它们之间的关系呢?



## 【例】食物链

并查集

河南省实验中学  
信息技术组

例题

程序自动分析

超市

打击犯罪

可爱的猴子

走廊泼水节

银河英雄传说

食物链

奇偶游戏

总结

边带权

扩展域

练习

- 如果  $x$  和  $y$  同类，那么它们与根结点之间的距离差满足  $d[x] - d[y] = 3k$ 。
- 如果  $x$  吃  $y$ ，那么它们与根结点之间的距离差满足  $d[x] - d[y] = 3k + 1$ 。
- 如果  $x$  被  $y$  吃，那么它们与根结点之间的距离差满足  $d[x] - d[y] = 3k + 2$ 。

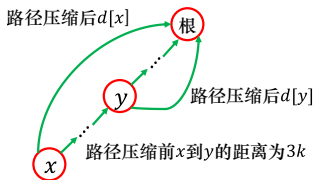


图:  $x$  和  $y$  是同类

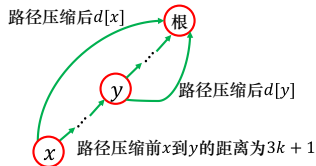


图:  $x$  吃  $y$

- 从上述结论可以看出，在路径压缩的同时让路径长度对 3 取余不会影响判定，同时为了避免判断  $d[x]$  和  $d[y]$  的大小，在所有对权值的操作中都需要对 3 取余。



## 【例】食物链

### 并查集

河南省实验中学  
信息技术组

### 例题

程序自动分析

超市

打击犯罪

可爱的猴子

走麻渡水节

银河英雄传说

食物链

奇偶游戏

### 总结

边带权

扩展域

### 练习

- 首先对于输入的  $x$  和  $y$ ，先判定其是否大于  $n$ 。
- 对于第一种说法，首先执行  $get(x)$  和  $get(y)$ 
  - 如果  $x$  和  $y$  在同一集合中，那么判断  $(d[x] - d[y])\%3$  是否等于 0；
  - 否则，合并  $x$  和  $y$  所属集合。
- 对于第二种说法，首先执行  $get(x)$  和  $get(y)$ 
  - 如果  $x$  和  $y$  在同一集合中，那么判断  $(d[x] - d[y] + 3)\%3$  是否等于 1；
  - 否则，合并  $x$  和  $y$  所属集合。

```
1 // 并查集
2 int get(int x)
3 {
4     if(x == f[x]) return x;
5     int fx = get(f[x]);
6     d[x] = (d[x] + d[f[x]]) % 3;
7     return f[x] = fx;
8 }
9 // 因为合并时需要更新 d 值 而且分类不同 所以此处不再实现 merge 函数
```



## 【例】食物链

### 并查集

河南省实验中学  
信息技术组

### 例题

程序自动分析  
超市  
打击犯罪  
可爱的猴子  
走廊泼水节  
银河英雄传说  
食物链  
奇偶游戏

### 总结

边带权  
扩展域

### 练习

```
1 // 第一种说法: x 和 y 是同类
2 int fx = get(x), fy = get(y);
3 if(fx == fy) // x 和 y 在同一个集合中
4 {
5     if((d[x] - d[y]) % 3 != 0) ++ans;
6 }
7 else // x 和 y 不在同一个集合中 合并两个集合
8 {
9     f[fy] = fx;
10    d[fy] = (d[x] - d[y] + 3) % 3;
11 }
```

- 如果把  $y$  所在的集合合并至  $x$  所在的集合, 则按照要求  $d[x] - (d[fy] + d[y]) = 3k$ , 所以  $d[x] - d[y] = 3k + d[fy]$

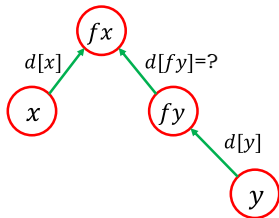


图:  $x$  和  $y$  是同类



## 【例】食物链

### 并查集

河南省实验中学  
信息技术组

### 例题

程序自动分析  
超市  
打击犯罪  
可爱的猴子  
走廊流水节  
银河英雄传说  
食物链  
奇偶游戏

### 总结

边带权  
扩展域

### 练习

```
1 // 第二种说法: x 吃 y
2 int fx = get(x), fy = get(y);
3 if(fx == fy) // x 和 y 在同一个集合中
4 {
5     if((d[x] - d[y] + 3) % 3 != 1) ++ans;
6 }
7 else // x 和 y 不在同一个集合中 合并两个集合
8 {
9     f[fy] = fx;
10    d[fy] = (d[x] - d[y] - 1 + 3) % 3;
11 }
```

- 如果把  $y$  所在的集合合并至  $x$  所在的集合, 则按照要求  $d[x] - (d[fy] + d[y]) = 3k + 1$ , 所以

$$d[x] - d[y] - 1 = 3k + d[fy]$$

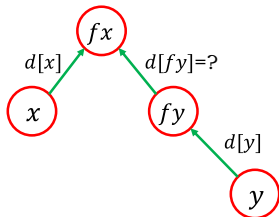


图:  $x$  吃  $y$



## 【例】食物链

### 并查集

河南省实验中学  
信息技术组

### 例题

程序自动分析  
超市  
打击犯罪  
可爱的猴子  
走廊泼水节  
银河英雄传说  
食物链  
奇偶游戏

### 总结

边带权  
扩展域

### 练习

- 方法 2: 扩展域并查集
- 把每个动物  $x$  拆成三个域, 同类域  $x_{self}$ 、捕食域  $x_{eat}$ 、天敌域  $x_{enemy}$ 。
- 对于第一种说法“ $x$  与  $y$  是同类”, 则说明  $x$  的同类和  $y$  的同类一样,  $x$  的捕食物种和  $y$  的捕食物种一样,  $x$  的天敌和  $y$  的天敌一样。此时, 合并  $x_{self}$  与  $y_{self}$ 、 $x_{eat}$  与  $y_{eat}$ 、 $x_{enemy}$  与  $y_{enemy}$  即可。
- 对于第二种说法“ $x$  吃  $y$ ”, 则说明  $x$  的捕食物种和  $y$  的同类一样,  $x$  的同类和  $y$  的天敌一样,  $x$  的天敌和  $y$  的捕食物种一样 ( $x$  吃  $y$ ,  $y$  吃  $z$ ,  $z$  吃  $x$ )。此时, 合并  $x_{eat}$  与  $y_{self}$ 、 $x_{self}$  与  $y_{enemy}$ 、 $x_{enemy}$  与  $y_{eat}$  即可。



## 【例】食物链

并查集

河南省实验中学  
信息技术组

例题

程序自动分析

超市

打击犯罪

可爱的猴子

走廊泼水节

银河英雄传说

食物链

奇偶游戏

总结

边带权

扩展域

练习

- 对于第一种说法“ $x$  与  $y$  是同类”，下面两种情况都能说明该说法为假话：
  - $x_{eat}$  与  $y_{self}$  在同一集合，说明  $x$  吃  $y$ ；
  - $x_{self}$  与  $y_{eat}$  在同一集合，说明  $y$  吃  $x$ ；

```
1 int f[3 * N];
2 for(int i = 1; i <= 3 * n; ++i) f[i] = i;
3 int x_self = get(x), x_eat = get(x + n), x_enemy = get(x + 2 * n);
4 int y_self = get(y), y_eat = get(y + n), y_enemy = get(y + 2 * n);
5 if(op == 1) // 说法 1: x 与 y 同类
6 {
7     // 如果 x 吃 y 或者 y 吃 x 说明是假话
8     if(x_eat == y_self || x_self == y_eat) ++ans;
9     else
10    {
11        f[y_self] = x_self; // x 和 y 是同类
12        f[y_eat] = x_eat; // x 的捕食物种和 y 的捕食物种是同类
13        f[y_enemy] = x_enemy; // x 的天敌和 y 的天敌是同类
14    }
15 }
```



## 【例】食物链

并查集

河南省实验中学  
信息技术组

例题

程序自动分析

超市

打击犯罪

可爱的猴子

走廊泼水节

银河英雄传说

食物链

奇偶游戏

总结

边带权

扩展域

练习

- 对于第二种说法“ $x$  吃  $y$ ”，下面两种情况都能说明该说法为假话：
  - $x_{self}$  与  $y_{self}$  在同一集合，说明  $x$  与  $y$  同类；
  - $x_{self}$  与  $y_{eat}$  在同一集合，说明  $y$  吃  $x$ ；

```
1 else if(op == 2) // 说法 2: x 吃 y
2 {
3     // 如果 x 和 y 是同类或者 y 吃 x 说明是假话
4     if(x_self == y_self || x_self == y_eat) ++ans;
5     else
6     {
7         f[y_self] = x_eat; // x 的捕食物种和 y 是同类
8         f[y_enemy] = x_self; // x 和 y 的天敌是同类
9         f[y_eat] = x_enemy; // x 的天敌和 y 的捕食物种是同类
10    }
11 }
```



## 【例】奇偶游戏

并查集

河南省实验中学  
信息技术组

例题

程序自动分析

超市

打击犯罪

可爱的猴子

走廊流水节

银河英雄传说

食物链

奇偶游戏

总结

边带权

扩展域

练习

小 A 和小 B 在玩一个游戏。首先，小 A 写了一个由 0 和 1 组成的序列  $s$ ，长度为  $n$ 。

然后，小 B 向小 A 提出了  $m$  个问题。在 each 问题中，小 B 指定两个数  $l, r$ ，小 A 回答  $s[l \sim r]$  中有奇数个 1 还是偶数个 1。

机智的小 B 发现小 A 有可能在撒谎。例如，小 A 曾经回答过  $s[1 \sim 3]$  中有奇数个 1， $s[4 \sim 6]$  中有偶数个 1，现在又回答  $s[1 \sim 6]$  中有偶数个 1，显然小 A 是自相矛盾的。请你帮助小 B 检查这  $m$  个答案，并指出在至少多少个回答之后可以确定小 A 一定在撒谎。即求出一个最小的  $k$ ，使得存在一个 01 序列满足第  $1 \sim k$  个回答，但不存在满足第  $1 \sim k + 1$  个回答的 01 序列。



## 【例】奇偶游戏

### 并查集

河南省实验中学  
信息技术组

### 例题

程序自动分析

超市

打击犯罪

可爱的猴子

走廊泼水节

银河英雄传说

食物链

奇偶游戏

### 总结

边带权

扩展域

### 练习

### 【输入格式】

第 1 行一个整数  $n(n \leq 10^9)$ ，表示序列的长度。

第 2 行一个整数  $m(m \leq 10^4)$ ，表示小 A 提出的问题。

接下来  $m$  行，每行包含两个整数  $l, r$  和回答， $l, r$  表示选择的子序列的起止位置，回答如果为 *even* 表示偶数，回答如果为 *odd* 表示奇数。

### 【输出格式】

一行一个整数，表示最小的  $k$ 。



## 【例】奇偶游戏

并查集

河南省实验中学  
信息技术组

例题

程序自动分析

超市

打击犯罪

可爱的猴子

走廊泼水节

银河英雄传说

食物链

奇偶游戏

总结

边带权

扩展域

练习

### 【样例 1 输入】

```
10
5
1 2 even
3 4 odd
5 6 even
1 6 even
7 10 odd
```

### 【样例 2 输入】

```
10
5
1 2 even
1 4 even
2 4 odd
1 10 even
3 10 even
```

### 【样例 1 输出】

```
3
```

### 【样例 2 输出】

```
5
```



## 【例】奇偶游戏

### 并查集

河南省实验中学  
信息技术组

### 例题

程序自动分析

超市

打击犯罪

可爱的猴子

走廊泼水节

银河英雄传说

食物链

奇偶游戏

### 总结

边带权

扩展域

### 练习

- 假设存在数组  $x[i]$  表示序列  $s[1 \sim i]$  的奇偶性，那么对于询问  $s[l \sim r]$ ：
  - 如果  $s[l \sim r]$  有偶数个 1，等价于  $x[l-1]$  与  $x[r]$  的奇偶性相同。
  - 如果  $s[l \sim r]$  有奇数个 1，等价于  $x[l-1]$  与  $x[r]$  的奇偶性不同。
- 于是问题转化为：有  $n+1$  个数字，确定它们  $x[0], x[1], \dots, x[n]$  的奇偶问题。



## 【例】奇偶游戏

并查集

河南省实验中学  
信息技术组

例题

程序自动分析  
超市  
打击犯罪  
可爱的猴子  
走廊泼水节  
银河英雄传说  
食物链  
奇偶游戏

总结

边带权  
扩展域

练习

- 方法 1: 边带权
- 定义  $d[x]$  表示  $x$  与  $f[x]$  的奇偶是否相同。 $d[x] = 0$  表示相同,  $d[x] = 1$  表示不同。
- 在路径压缩时, 对  $x$  到根结点上所有边权做异或运算, 即可得到  $x$  与根结点的奇偶关系。
- 对于第  $i$  个问题  $(l[i], r[i], c[i])$ , 它的  $l[i] - 1$  和  $r[i]$  在离散化后分别为  $x, y$ ,  $c[i] = 0$  表示奇数,  $c[i] = 1$  表示偶数。

```
1 for(int i = 1; i <= m; ++i)
2   a[i] = l[i] - 1, a[i + m] = r[i];
3 sort(a + 1, a + 2 * m + 1);
4 n = unique(a + 1, a + 2 * m + 1) - (a + 1);
```



## 【例】奇偶游戏

### 并查集

河南省实验中学  
信息技术组

### 例题

程序自动分析  
超市  
打击犯罪  
可爱的猴子  
走廊泼水节  
银河英雄传说  
食物链  
奇偶游戏

### 总结

边带权  
扩展域

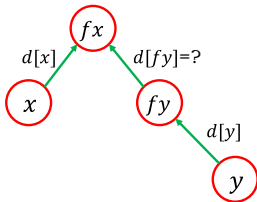
### 练习

- 首先检查  $x, y$  是否在同一个集合内，在执行  $get(x)$  和  $get(y)$  时会有路径压缩。
- 若  $x, y$  在同一个集合内，但是路径压缩后  $d[x] \oplus d[y] \neq c[i]$ ，说明在撒谎。
- 若  $x, y$  不在同一个集合内，设两个集合的根结点为  $fx, fy$ ，并令  $fy$  合并到  $fx$ ，那么显然有

$$c[i] = d[x] \oplus (d[y] \oplus d[fy])$$

故而

$$d[fy] = d[x] \oplus d[y] \oplus c[i]$$





## 【例】奇偶游戏

并查集

河南省实验中学  
信息技术组

例题

程序自动分析

超市

打击犯罪

可爱的猴子

走廊泼水节

银河英雄传说

食物链

奇偶游戏

总结

边带权

扩展域

练习

```
1 int i = 1;
2 for(i = 1; i <= m; ++i)
3 {
4     int x = lower_bound(a + 1, a + n + 1, l[i] - 1) - a;
5     int y = lower_bound(a + 1, a + n + 1, r[i]) - a;
6     int fx = get(x), fy = get(y);
7     if(fx == fy)
8     {
9         // 奇偶性与给的答案矛盾
10        if (d[x] ^ d[y] != c[i]) break;
11    }
12    else
13    {
14        f[fy] = fx;
15        // c[i] = d[x] ⊕ (d[y] ⊕ d[fy])
16        d[fy] = d[x] ^ d[y] ^ c[i];
17    }
18 }
```



## 【例】奇偶游戏

### 并查集

河南省实验中学  
信息技术组

### 例题

程序自动分析

超市

打击犯罪

可爱的猴子

走廊泼水节

银河英雄传说

食物链

奇偶游戏

### 总结

边带权

扩展域

### 练习

- 方法 2: 扩展域
- 将每个变量拆分成两个结点  $x_{odd}$  和  $x_{even}$ , 分别表示奇数域和偶数域。
- 对于第  $i$  个问题  $(l[i], r[i], c[i])$ , 它的  $l[i] - 1$  和  $r[i]$  在离散化后分别为  $x, y$ ,  $c[i] = 0$  表示奇数,  $c[i] = 1$  表示偶数。



## 【例】奇偶游戏

### 并查集

河南省实验中学  
信息技术组

### 例题

程序自动分析

超市

打击犯罪

可爱的猴子

走廊泼水节

银河英雄传说

食物链

奇偶游戏

### 总结

边带权

扩展域

### 练习

- 若  $x$  和  $y$  的奇偶性不同：
  - 首先检查  $x$  和  $y$  是否在同一个集合，如果在同一个集合，那么判定撒谎。
  - 否则，就合并  $x_{odd}$  和  $y_{even}$ 、 $x_{even}$  和  $y_{odd}$ 。

```
1 int f[2 * N];
2 int fx = get(x), sx = get(x + n); // xodd xeven
3 int fy = get(y), sy = get(y + n); // yodd yeven
4 if(c[i]) // x 和 y 的奇偶性不同
5 {
6     if(fx == fy || sx == sy) break;
7     else merge(fx, sy), merge(fy, sx);
8 }
```



## 【例】奇偶游戏

### 并查集

河南省实验中学  
信息技术组

### 例题

程序自动分析  
超市  
打击犯罪  
可爱的猴子  
走麻渡水节  
银河英雄传说  
食物链  
奇偶游戏

### 总结

边带权  
扩展域

### 练习

- 若  $x$  和  $y$  的奇偶性相同：
  - 首先检查  $x$  和  $y$  是否在同一个集合，如果不在在同一个集合，那么判定撒谎。
  - 否则，就合并  $x_{odd}$  和  $y_{odd}$ 、 $x_{even}$  和  $y_{even}$ 。

```
1 int fx = get(x), sx = get(x + n); // xodd xeven
2 int fy = get(y), sy = get(y + n); // yodd yeven
3 if(!c[i]) // x 和 y 的奇偶性不同
4 {
5     if(fx == sy || fy == sx) break;
6     else merge(fx, fy), merge(sx, sy);
7 }
```



# 边带权并查集

## 并查集

河南省实验中学  
信息技术组

## 例题

程序自动分析  
超市  
打击犯罪  
可爱的猴子  
走廊泼水节  
银河英雄传说  
食物链  
奇偶游戏

## 总结

边带权  
扩展域

## 练习

- 并查集树形结构中每个结点除了记录父结点外，还要记录到父结点的距离，该距离称为权值。
- 权值的主要意义就是将已经在同一集合中的点进行分类，根据权值不同表示集合内点的关系。
- 在查询的过程中，会使用路径压缩，每个访问过的结点都会直接指向根结点。所以我们需要更新结点记录的距离值(父节点发生变化)，在此过程中可以得到结点到根结点路径上的信息。
- 实现：维护一个数组  $d$ ，用  $d[x]$  存储结点  $x$  到父结点  $f[x]$  之间的边权。

```
1 int get(int x)
2 {
3     if(x == f[x]) return x;
4     int fx = get(f[x]);
5     d[x] = d[x] + d[f[x]];
6     return f[x] = fx;
7 }
```



# 扩展域并查集

## 并查集

河南省实验中学  
信息技术组

## 例题

程序自动分析  
超市  
打击犯罪  
可爱的猴子  
走廊泼水节  
银河英雄传说  
食物链  
奇偶游戏

## 总结

边带权  
扩展域

## 练习

- 扩展域并查集解决了一种多个有相互关系的并查集，放在一起考虑的问题。一般的并查集应用一般就是判断在不在一个集合，拓展域并查集讲的是多个集合，之间有相互关系（一般为相互排斥关系），判断是否在一个集合等。
- 对与最简单的并查集来说，如果两个是同一类，合并即可，但是如果两个相互对立呢？首先想法就是建立两个并查集，但是怎么把两个并查集联系起来呢？扩展域。
- 何谓扩展域？把一个个体拆成多个，比方说两个集合存在对立关系，那么对于一个个体  $x$ ，我们假设存在一个个体  $x+n$ ， $x$  和  $x+n$  这两个是处于对立关系的<sup>1</sup>
  - 当我们说  $x$  和  $y$  是同类的时候，那么也就是说  $x$  和  $y$  属于一个并查集，且  $x+n$  和  $y+n$  属于一个并查集；
  - 当我们说  $x$  和  $y$  对立的时候，意思就是在说： $x+n$  和  $y$  在同一并查集， $x$  和  $y+n$  在同一并查集。

<sup>1</sup>敌人的敌人是朋友，朋友的敌人是敌人。



# 练习

## 并查集

河南省实验中学  
信息技术组

## 例题

程序自动分析

超市

打击犯罪

可爱的猴子

走廊泼水节

银河英雄传说

食物链

奇偶游戏

## 总结

边带权

扩展域

## 练习

- 破译密文 (COGS 20)
- 打击犯罪 (COGS 3997)
- 可爱的猴子 (COGS 2043)
- 程序自动分析 [NOI 2015](COGS 2018)
- 黑魔法师之门 (COGS 1015)
- 超市 (COGS 3097)
- 搭配购买 (COGS 1571)
- 批发市场 (COGS 4022)
- 机器人搬运 (COGS 1670)
- 走廊泼水节 (COGS 3091)
- 整数合并 (COGS 487)



# 练习

## 并查集

河南省实验中学  
信息技术组

## 例题

程序自动分析  
超市  
打击犯罪  
可爱的猴子  
走廊泼水节  
银河英雄传说  
食物链  
奇偶游戏

## 总结

边带权  
扩展域

## 练习

- 银河英雄传说 [NOI 2002](COGS 260)
- 关押罪犯 [NOIP 2010](COGS 520)
- 犯罪团伙 (COGS 2328)
- 奇偶性游戏 (COGS 1718)
- 食物链 [NOI 2001](COGS 298)
- 剪刀石头布 (COGS 3540)
- 真正的说谎者 (COGS 3541)