



并查集

河南省实验中学
信息技术组

引例

概念

实现

简单实现
路径压缩
启发式合并

例题

社交网络
程序自动分析
打击犯罪
岛屿
搭配购买

小结

练习

并查集

Disjoint Set

河南省实验中学信息技术组

2026年05月09日



【例】社交网络

并查集

河南省实验中学
信息技术组

引言

概念

实现

简单实现
路径压缩
启发式合并

例题

社交网络
程序自动分析
打击犯罪
岛国
搭配购买

小结

练习

【题目描述】

你知道吗？你和任何一个陌生人之间所间隔的人不会超六个，也就是说，最多通过六个人你就能够认识任何一个陌生人。

现在给定 n 个人，假设这 n 个人开始时互相都不认识，现在逐步令他们直接互相认识，当然中间也会询问是两个人是否能有联系。

【输入格式】

第一行两个整数 n, m ($2 \leq n \leq 2 \times 10^4, m \leq 10^5$)，表示总共有 n 个人，有 m 次操作。

接下来 m 行，每行三个整数 op, x, y 。

若 $op = 0$ ，则说明 x 和 y 之间直接认识。

若 $op = 1$ ，根据当前已知的关系，询问 x 和 y 是否有联系。

【输出格式】

对于每个询问，若 x 和 y 有联系，则输出 Yes，否则输出 No。



【引例】社交网络

并查集

河南省实验中学
信息技术组

引例

概念

实现

简单实现
路径压缩
启发式合并

例题

社交网络
程序自动分析
打击犯罪
岛国
搭配购买

小结

练习

【样例输入】

```
9 10
0 2 4
0 5 7
0 1 3
0 1 2
1 3 4
0 8 9
0 5 6
1 8 5
0 2 3
1 7 6
```

【样例输出】

```
Yes
No
Yes
```



【引例】社交网络

并查集

河南省实验中学
信息技术组

引例

概念

实现

简单实现
路径压缩
启发式合并

例题

社交网络
程序自动分析
打击犯罪
岛国
搭配购买

小结

练习

- 将每个人抽象为一个点，如果两个人直接认识，那么就在他们之间连一条无向边，很容易得出一个有 n 个顶点的无向图。
- 显然，在同一个连通块中的所有顶点都有联系。只需要利用 BFS/DFS 求出连通块，然后对于任意提问，只需要判断两个顶点是否在同一个连通块中即可。
- 时间复杂度为 $O(M(N + M))$ ，其中 M 为新增关系数。



【引例】社交网络

并查集

河南省实验中学
信息技术组

引例

概念

实现

简单实现
路径压缩
启发式合并

例题

社交网络
程序自动分析
打击犯罪
岛国
搭配购买

小结

练习

- 对于两个人，无需考虑他们的具体的关系，只需考虑他们**是否在**同一个连通块中。
- 因此可以使用集合，所有人都属于某一个集合，当给出两个在不同集合中人存在亲戚关系时，合并两个集合——这就是**并查集**。



并查集

并查集

河南省实验中学
信息技术组

引例

概念

实现

简单实现
路径压缩
启发式合并

例题

社交网络
程序自动分析
打击犯罪
岛国
搭配购买

小结

练习

并查集 (Disjoint Set) 是一种可以动态维护若干个不重叠的集合，并支持**合并**与**查询**的数据结构。并查集一般包含如下两种操作：

- get: 查询某一个元素所在集合。
- merge: 合并两个集合。



实现

并查集

河南省实验中学
信息技术组

引例

概念

实现

简单实现

路径压缩

启发式合并

例题

社交网络

程序自动分析

打击犯罪

岛国

搭配购买

小结

练习

设所有集合中共有 n 个元素，可以用 $1 \sim n$ 这 n 个整数来表示。为了实现并查集，需要解决两个问题：

- 如何表示集合？为每个集合选择一个固定的元素，作为整个集合的代表。
- 如何表示元素所属集合？定义 $f[x]$ 表示元素 x 所在集合的代表。



实现 1：数组实现

并查集

河南省实验中学
信息技术组

引例

概念

实现

简单实现
路径压缩
启发式合并

例题

社交网络
程序自动分析
打击犯罪
岛国
搭配购买

小结

练习

- 查询：对于元素 x ，直接返回 $f[x]$ 即可。
- 合并：对于集合 A 和集合 B，合并时需将某一个集合中的所有元素中存储的代表元全部修改为另一个集合的代表元。

```
1 void init() // 初始时每个元素自成一个集合
2 {
3     for(int i = 1; i <= n; ++i) f[i] = i;
4 }
5 int get(int x)
6 {
7     return f[x];
8 }
9 void merge(int x, int y)
10 {
11     int fx = get(x), fy = get(y);
12     if(fx == fy) return;
13     for(int i = 1; i <= n; ++i) if(f[i] == fy) f[i] = fx;
14 }
```



实现 1：数组方式

并查集

河南省实验中学
信息技术组

引例

概念

实现

简单实现
路径压缩
启发式合并

例题

社交网络
程序自动分析
打击犯罪
岛屿

小结

练习

假设集合如下所示：

元素	1	2	3	4	5	6	7	8	9	10
f	1	2	2	2	2	2	2	2	2	2

然后执行 $merge(1, 5)$

元素	1	2	3	4	5	6	7	8	9	10
f	1	2→1	2→1	2→1	2→1	2→1	2→1	2→1	2→1	2→1

上述操作的时间复杂度为 $O(N)$ ，如果多次合并都出现这种情况，那么将会花费大量时间。这是因为每个顶点存储了自己所在集合编号，互相之间没有关联，这样修改时只能一一修改。有没有更好的方法呢？



实现 2：树形实现

并查集

河南省实验中学
信息技术组

引例

概念

实现

简单实现
路径压缩
启发式合并

例题

社交网络
程序自动分析
打击犯罪
岛屿
搭配购买

小结

练习

- 每个子集合用一棵树来表示，树中的每个顶点表示集合中的元素，树根是集合的代表。
- 定义 $f[x]$ 表示元素 x 的父结点。
- 查询：沿着父结点方向向根结点访问，直到到达树根。
- 合并：令其中一个根为另一个根的子结点即可。



树形实现

并查集

河南省实验中学
信息技术组

引例

概念

实现

简单实现
路径压缩
启发式合并

例题

社交网络
程序自动分析
打击犯罪
岛屿
搭配购买

小结

练习

```
1 void init() // 初始时每个元素自成一个集合
2 {
3     for(int i = 1; i <= n; ++i) f[i] = i;
4 }
5 int get(int x) //非递归
6 {
7     while(x != f[x]) x = f[x];
8     return x;
9 }
10 int get(int x) //递归
11 {
12     if(x == f[x]) return x;
13     return get(f[x]);
14 }
15 void merge(int x, int y)
16 {
17     int fx = get(x), fy = get(y);
18     if(fx != fy) f[fy] = fx;
19 }
```



实现 2：树形实现

并查集

河南省实验中学
信息技术组

引例

概念

实现

简单实现
路径压缩
启发式合并

例题

社交网络
程序自动分析
打击犯罪
岛国
搭配购买

小结

练习

下标	1	2	3	4	5	6	7	8	9
$f[x]$	2	2	2	9	9	1	3	2	9

下标	1	2	3	4	5	6	7	8	9
$f[x]$	2	9	2	9	9	1	3	2	9

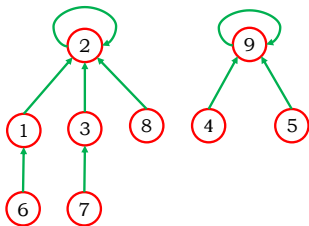


图: merge(5, 6) 前

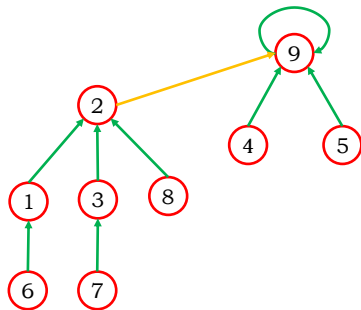


图: merge(5, 6) 后



实现 2：树形实现

并查集

河南省实验中学
信息技术组

引例

概念

实现

简单实现
路径压缩
启发式合并

例题

社交网络
程序自动分析
打击犯罪
岛国
搭配购买

小结

练习

- 虽然查询步骤略多，但是合并速度变快。
- 但是如果合并后树的深度越来越深，那么查询一次将会很费时。最坏的情况：集合树退化成一条单链，此时查询操作的时间复杂度是 $O(N)$ ，这在数据量大时是不能接受的。
- 通过优化，可以解决上述问题，一般并查集使用树形实现。



路径压缩

并查集

河南省实验中学
信息技术组

引例

概念

实现

简单实现
路径压缩
启发式合并

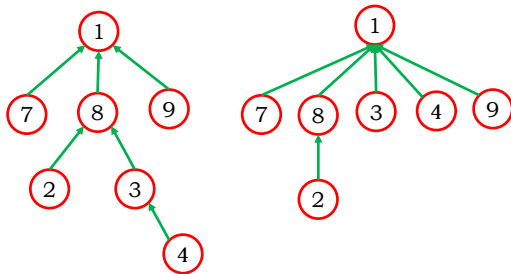
例题

社交网络
程序自动分析
打击犯罪
岛屿
搭配购买

小结

练习

- 在树形结构中，只需知道元素的根结点，而并不关心这棵树的具体形态。
- 在执行查询操作的同时，将访问过的每个结点(祖先)都直接指向根结点，这就是路径压缩。
- 例如对于如下集合，执行 `get(4)` 后：



图：执行 `get(4)` 的路径压缩



路径压缩

并查集

河南省实验中学
信息技术组

引例

概念

实现

简单实现
路径压缩
启发式合并

例题

社交网络
程序自动分析
打击犯罪
岛国
搭配购买

小结

练习

```
1 void init()      // 初始时每个元素自成一个集合
2 {
3     for(int i = 1; i <= n; ++i) f[i] = i;
4 }
5
6 int get(int x)
7 {
8     if(x == f[x]) return x;
9     // return get(f[x]);
10    return f[x] = get(f[x]);    // 路径压缩
11 }
12
13 void merge(int x, int y)
14 {
15     int fx = get(x), fy = get(y);
16     if(fx != fy) f[fy] = fx;
17 }
```



启发式合并

并查集

河南省实验中学
信息技术组

引例

概念

实现

简单实现
路径压缩
启发式合并

例题

社交网络
程序自动分析
打击犯罪
岛屿
搭配购买

小结

练习

- 优化的目的是减少元素的平均查询时间 (结点平均层次)。
- 所以在合并时, 可以将“小”的树合并至“大”的树, 这就是启发式合并。
- 例如: 可以将元素个数少的集合合并至元素个数多的集合; 将树高度小的集合合并至树高度大的集合。



启发式合并

并查集

河南省实验中学
信息技术组

引例

概念

实现

简单实现
路径压缩
启发式合并

例题

社交网络
程序自动分析
打击犯罪
岛国
搭配购买

小结

练习

```
1 int f[N];
2 int d[N]; // d[i] 表示以 i 为根的树的大小
3 // 以集合元素个数作为合并参考
4 void init(); // 初始时每个元素自成一个集合
5 int get(int x);
6 void merge(int x, int y)
7 {
8     int fx = get(x), fy = get(y);
9     if(fx != fy)
10    {
11        // 集合 fx 元素个数比集合 fy 多
12        if(d[fx] >= d[fy]) f[fy] = fx, d[fx] += d[fy];
13        else f[fx] = fy, d[fy] += d[fx];
14    }
15 }
```



【例】社交网络

并查集

河南省实验中学
信息技术组

引例

概念

实现

简单实现
路径压缩
启发式合并

例题

社交网络
程序自动分析
打击犯罪
岛屿
搭配购买

小结

练习

- 初始化：每个人独自成为一个集合。
- 亲戚关系：对于给定的两人直接认识，合并两个人所在集合。
- 询问：对于每个询问，分别查询两人所在集合，如果相同，输出 Yes，否则，输出 No。

```
1 int n, m; cin >> n >> m;  
2 init();  
3 while(m--)  
4 {  
5     int op, x, y; cin >> op >> x >> y;  
6     if(op == 0) merge(x, y);    // x 直接认识 y  
7     else if(op == 1)           // 查询  
8     {  
9         if(get(x) == get(y)) cout << "Yes\n";  
10        else cout << "No\n";  
11    }  
12 }
```



【例】程序自动分析

并查集

河南省实验中学
信息技术组

引言

概念

实现

简单实现
路径压缩
启发式合并

例题

社交网络
程序自动分析
打击犯罪
岛国
搭配购买

小结

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

引例

概念

实现

简单实现
路径压缩
启发式合并

例题

社交网络
程序自动分析
打击犯罪
岛屿
搭配购买

小结

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

引例

概念

实现

简单实现
路径压缩
启发式合并

例题

社交网络
程序自动分析
打击犯罪
岛国
搭配购买

小结

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

引例

概念

实现

简单实现
路径压缩
启发式合并

例题

社交网络
程序自动分析
打击犯罪
岛屿
搭配购买

小结

练习

- 显然对于所有相等的约束条件，可以利用并查集合并将所有相等的元素加入同一个集合。
- 然后对于不相等的约束条件，查询是否属于不同的集合即可。

```
1 for(int i = 1; i <= n; ++i) cin >> a[i] >> b[i] >> c[i];
2 for(int i = 1; i <= 10000; ++i) f[i] = i;
3 for(int i = 1; i <= n; ++i) // 相等 约束条件
4     if(c[i]) merge(a[i], b[i]);
5 bool flag = true;
6 for(int i = 1; i <= n; ++i) // 不相等约束条件
7 {
8     if(c[i]) continue;
9     if(get(a[i]) == get(b[i])) { flag = false; break; }
10 }
```



【例】程序自动分析

并查集

河南省实验中学
信息技术组

引例

概念

实现

简单实现
路径压缩
启发式合并

例题

社交网络
程序自动分析
打击犯罪
岛国
搭配购买

小结

练习

- 并查集的下标范围过大，需要对输入的 i, j 进行离散化。
- 时间复杂度： $O(TN \log N)$ 。

```
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】打击犯罪

并查集

河南省实验中学
信息技术组

引例

概念

实现

简单实现
路径压缩
启发式合并

例题

社交网络
程序自动分析
打击犯罪
岛屿
搭配购买

小结

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

引例

概念

实现

简单实现
路径压缩
启发式合并

例题

社交网络
程序自动分析
打击犯罪
岛屿
搭配购买

小结

练习

【样例输入】

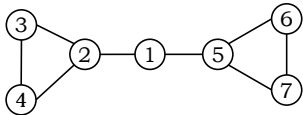
```
7
2 2 5
3 1 3 4
2 2 4
2 2 3
3 1 6 7
2 5 7
2 5 6
```

【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

引例

概念

实现

简单实现
路径压缩
启发式合并

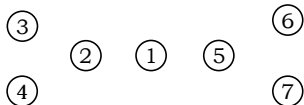
例题

社交网络
程序自动分析
打击犯罪
岛屿
搭配购买

小结

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

引例

概念

实现

简单实现
路径压缩
启发式合并

例题

社交网络
程序自动分析
打击犯罪
岛国
搭配购买

小结

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】岛国

并查集

河南省实验中学
信息技术组

引例

概念

实现

简单实现
路径压缩
启发式合并

例题

社交网络
程序自动分析
打击犯罪
岛国
搭配购买

小结

练习

【题目描述】

有一个岛国的领地是一块坐标从 $(1,1)$ 到 (K,K) 的正方形 (包括领海和领陆, 坐标 (x,y) 是指 (x,y) 这块土地, 并非一个点)。

卫星信息会告诉你这个国家的土地情况, 希望你能根据给出的信息计算出这个国家有多少个岛。

卫星给出的信息形如 x_1, y_1, x_2, y_2 , 表示左下角坐标为 (x_1, y_1) , 右上角坐标为 (x_2, y_2) 的这一个矩形区域是陆地。

【输入格式】

第一行一个整数 n , 表示卫星会传送给你 n 条信息。

接下来 n 行每行有 4 个整数, x_1, y_1, x_2, y_2 。

【输出格式】

共一行, 包含一个整数, 表示这个国家的岛的数量 (两块陆地有一个公共顶点不算一个区域)。

【数据规模与约定】

对于 30% 的数据: $K \leq 1000, n \leq 100$ 。

对于 100% 的数据: $K \leq 2 \times 10^4, n \leq 5000, 1 \leq x_1 \leq x_2 \leq K, 1 \leq y_1 \leq y_2 \leq K$ 。



【例】岛国

并查集

河南省实验中学
信息技术组

引例

概念

实现

简单实现
路径压缩
启发式合并

例题

社交网络
程序自动分析
打击犯罪
岛国
搭配购买

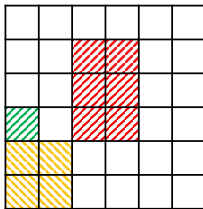
小结

练习

【样例输入】

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

【样例解释】



【样例输出】

```
2
```



【例】岛国

并查集

河南省实验中学
信息技术组

引例

概念

实现

简单实现
路径压缩
启发式合并

例题

社交网络
程序自动分析
打击犯罪
岛国
搭配购买

小结

练习

- 将每块陆地覆盖的区域在地图上标记，然后用 BFS/DFS 查找四连通分量，这样无论是空间还是时间都无法接受。
- 将每块陆地看成一个顶点，问题变为判断这 n 个顶点的连通块的个数，利用并查集处理后集合的个数即为连通块的个数。
- 当两块陆地重合时，合并两个顶点的集和。

```
1 for(int i = 1; i <= n; ++i)
2     for(int j = i + 1; j <= n; ++j)
3         if(get(i) != get(j) && conn(i, j)) merge(i, j);
4 int ans = 0;
5 for(int i = 1; i <= n; ++i)
6     if(get(i) == i) ++ans; //集合的个数
```



【例】岛国

并查集

河南省实验中学
信息技术组

引例

概念

实现

简单实现
路径压缩
启发式合并

例题

社交网络
程序自动分析
打击犯罪
岛国
搭配购买

小结

练习

- 如何判断两块陆地是否相连？

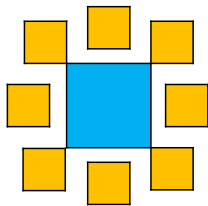


图: 不相连

```
1 bool conn(int i, int j)
2 {
3     if(a[i].y2 + 1 < a[j].y1 || a[j].y2 + 1 < a[i].y1 ||
4         a[i].x2 + 1 < a[j].x1 || a[j].x2 + 1 < a[i].x1)
5         return false; // 不相邻
6     if(a[i].x2 + 1 == a[j].x1 && (a[i].y1 == a[j].y2 + 1 || a[i].y2 + 1 == a[j].y1) ||
7         a[j].x2 + 1 == a[i].x1 && (a[j].y1 == a[i].y2 + 1 || a[j].y2 + 1 == a[i].y1))
8         return false; // 只有一个公共点
9     return true;
10 }
```



【例】搭配购买

并查集

河南省实验中学
信息技术组

引例

概念

实现

简单实现
路径压缩
启发式合并

例题

社交网络
程序自动分析
打击犯罪
岛屿
搭配购买

小结

练习

【题目描述】

Joe 觉得云朵很美，决定去山上的商店买一些云朵，商店里有 n 朵云，云朵被编号为 $1, 2, \dots, n$ ，并且每朵云都有一个价值。但是商店老板跟他说，一些云朵要搭配来买才好，所以买一朵云则与这朵云有搭配的云都要买，但是 *Joe* 的钱有限，所以他希望买的价值越多越好。

【输入格式】

第 1 行三个整数 n, m, w 表示 n 朵云， m 个搭配，*Joe* 有 w 的钱。

接下来 n 行，每行两个整数 c_i, d_i 表示 i 朵云的价钱和价值。

接下来 m 行，每行两个整数 u_i, v_i 表示买 u_i 必须买 v_i ，同理，如果买 v_i 就必须买 u_i 。

【输出格式】

一行，表示可以获得的最大价值。

【数据范围】

$1 \leq n, w, c_i, d_i \leq 10^4$ ， $0 \leq m \leq 5 \times 10^3$



【例】搭配购买

并查集

河南省实验中学
信息技术组

引例

概念

实现

简单实现
路径压缩
启发式合并

例题

社交网络
程序自动分析
打击犯罪
岛国
搭配购买

小结

练习

【样例输入】

```
5 3 10
3 10
3 10
3 10
5 100
10 1
1 3
3 2
4 2
```

【样例输出】

```
1
```



【例】搭配购买

并查集

河南省实验中学
信息技术组

引例

概念

实现

简单实现
路径压缩
启发式合并

例题

社交网络
程序自动分析
打击犯罪
岛屿
搭配购买

小结

练习

- 如果不考虑搭配购买的问题，问题显然是一个 01 背包模型。
- 考虑搭配购买，所有需要搭配购买的商品需要同时购买，那么就相当于购买一个价钱和价值为搭配的商品的价钱价值的新商品。
- 可以用并查集来处理，将所有需要搭配购买的商品合并到一个集合，同时将它们的价钱和价值合并到根结点，然后只对根结点的商品进行 01 背包即可。

```
1 for(int i = 1; i <= m ; ++i)
2 {
3     int x, y; cin >> x >> y;
4     merge(x, y);
5 }
6 for(int i = 1; i <= n; ++i)
7 {
8     if(i != f[i]) continue; // 只能是根结点代表的商品
9     for(int j = w; j >= c[i]; --j)
10         g[j] = max(g[j], g[j - c[i]] + d[i]);
11 }
12 cout << g[w];
```



小结

并查集

河南省实验中学
信息技术组

引例

概念

实现

简单实现
路径压缩
启发式合并

例题

社交网络
程序自动分析
打击犯罪
岛屿
搭配购买

小结

练习

- 并查集实现时采用树形结构。
- 通过路径压缩和启发式合并可以大大提高效率，单次操作的时间复杂度为 $O(k)$ ， k 是一个较小的数字。
- 由于有路径压缩的存在，所以每棵树被合并时高度不会很大，所以进行启发式合并意义不是很大；而且启发式合并需要额外的空间来记录集合“大小”。
- 在某些问题中，集合“大小”的信息有额外的用处。
- 并查集在诸多方面都有应用，可以大大加快其他算法的速度。
 - 等价类划分
 - 无向图连通分量
 - 判环
 - Kruskal 算法
 - 最近公共祖先



并查集标准模板

并查集

河南省实验中学
信息技术组

引例

概念

实现

简单实现
路径压缩
启发式合并

例题

社交网络
程序自动分析
打击犯罪
岛国
搭配购买

小结

练习

```
1 void init() // 初始时每个元素自成一个集合
2 {
3     for(int i = 1; i <= n; ++i) f[i] = i;
4 }
5
6 int get(int x)
7 {
8     if(x == f[x]) return x;
9     return f[x] = get(f[x]);
10 }
11
12 void merge(int x, int y)
13 {
14     int fx = get(x), fy = get(y);
15     if(fx != fy) f[fy] = fx;
16 }
```



练习

并查集

河南省实验中学
信息技术组

引例

概念

实现

简单实现
路径压缩
启发式合并

例题

社交网络
程序自动分析
打击犯罪
岛国
搭配购买

小结

练习

- 社交网络 (COGS 1678)
- 亲戚 (COGS 259)
- 朋友圈 (COGS 3008)
- 程序自动分析 [NOI 2015](COGS 2018)
- 打击犯罪 (COGS 3997)
- 岛国 (COGS 456)
- 相交的矩形 (COGS 234)
- 黑魔法师之门 (COGS 1015)
- 破译密文 [HAOI 2005](COGS 20)
- 搭配购买 (COGS 1571)
- 批发市场 (COGS 4022)
- 机器人搬运 (COGS 1670)
- 可爱的猴子 (COGS 2043)