小马加编信息学教案(四十)

并查集

一. 课程内容

1. 并查集的介绍
2. 并查集的基本思路

二. 知识讲解

1. 并查集的介绍

> 并查集从名字可以看出，主要涉及两种基本操作，对集合的**合并**和**查找**。

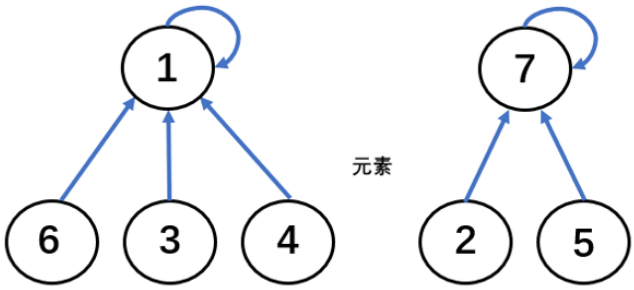
> 加入现在有n个元素，一开始可以视为每个元素都为独立的一个集合。有两种操作：

- 合并：给定x, y，将x, y所处的集合合并。
- 查询：给定x, y，判断他们是否在同一个结合。

2. 并查集的基本思路

2. 1 思路一

> 一种简单的思路就是对于每个集合可以规定一个关键点，并令$key[i]$表示第i个元素所处集合的关键点，一开始每个元素都为独立的一个集合，所以`key[i] = i`。



箭头指向的为key数组中存的值

> 考虑如何处理合并和查找操作。

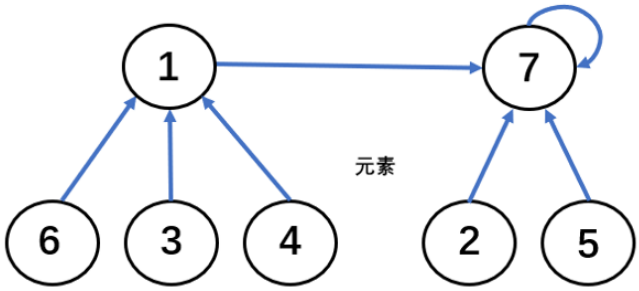
- 查找：这个很好理解，直接判断关键点是否相同即可。对于元素x,y，若`key[x] == key[y]`，则x,y在同一个集合，否则不在。
- 合并：那么相当于将x所处集合所有元素的关键点变为$key[y]$，即将x所处集合中所有元素加到y集合。

> 由于存在合并操作，需要遍历所有key值等于$key[x]$的元素，所以时间复杂度是$O(n^2)$的。

2. 2 思路二

> 思路一中，每次进行合并操作都需要遍历整个数组，考虑怎么优化。

> 考虑将思路一中的元素3和元素5所处的集合合并，我们尝试只修改元素1的key值。令`key[1] = 7`



箭头指向的为key数组中存的值

> 那么查找时就不能直接比较key值是否相同，但是换个方式思考，其实key值的指向就是合并集合的过程，一个集合中所有的元素一直沿着箭直跳，肯定会跳到同一个元素，即为当前集合的关键点，那么我们判断x,y是否在同一个集合时，只需判断他们的关键点是否相同即可。

```
//获取集合的关键点
int get(int x) {
        if (key[x] == x) return x;
        return get(key[x]);
}
```

```
        scanf("%d%d", &x, &y);
        if (get(x) == get(y)) printf("YES\n"); else printf("NO\n");
        // 若x,y属于一个集合则输出"YES"，否则输出"NO"。
```

> 合并x,y属于的两个集合时，只需要把x集合的关键点指向y集合的关键点即可。即

```
        key[get(x)] = get(y);
```

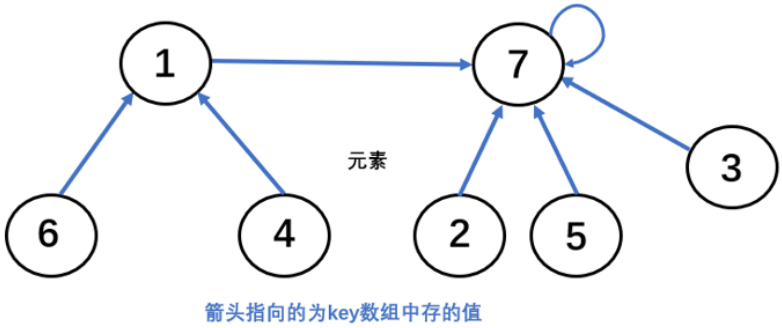
> 但是每次执行get函数都可能向上跳n次。所以总的复杂度还是$O(n^2)$。

2. 3 思路三

> 那么继续对思路二进行优化，思路二主要寻找关键点时要不断往key值方向跳。其实每次我们访问了$get(x)$后就可以将$key[x]$修改成$get(x)$，减少很多不必要的操作，大大优化时间复杂度。

```
int get(int x) {
        if (key[x] == x) return x;
        key[x] = get(key[x]);
        return key[x];
}
```

> 同样，将思路一中元素3和元素5按升级后的方法进行合并后，就会变成下图。



箭头指向的为key数组中存的值

> 对于合并操作仍然采用上面思路二的方法。

```
        if (get(x) != get(y)) key[get(x)] = get(y);
        //如果get(x) == get(y)则说明x和y已经在一个集合了。
```

> 用这种方法，时间复杂度就可以优化到$O(n\alpha(n))$，$\alpha(n)$可以视为常数，所以可以近似为时间复杂度是$O(n)$。

> 思路三就是我们常说的并查集算法，能较快的对集合进行合并和判断两个元素是否再同一个集合。

三. 经典例题

1. 并查集

> 加入现在有n个元素，一开始可以视为每个元素都为独立的一个集合。有两种操作：
>
> - 合并：给定x, y，将x, y所处的集合合并。
> - 查询：给定x, y，判断他们是否在同一个结合。
>
> 现在总共有m个操作。

> 输入格式：
> 第一行两个整数n, m。
> 接下来m个操作，每个操作包括三个整数z_i, x_i, y_1。若$z_i = 1$则将x_i, y_i所在集合合并。若$z_i = 2$则输出x_i与y_i是否在同一集合内，若是输出Y，否则输出N。
> $n \leq 10^4$
> $m \leq 2 * 10^5$

> 输出格式：
> 对于每个查询操作输出Y或N。

样例输入	样例输出
4 7 2 1 2 1 1 2 2 1 2 1 3 4 2 1 4 1 2 3 2 1 4	 N Y N Y

2. 村村通

> 某市调查城镇交通状况，得到现有城镇道路统计表。表中列出了每条道路直接连通的城镇。市政府“村村通工程”的目标是使全市任何两个城镇间都可以实现交通（但不一定有直接的道路相连，只要相互之间可达即可）。请你计算出最少还需要建设多少条道路?

> 输入格式:
> 每个输入文件包含若干组测试测试数据。
> 每组测试数据的第一行给出两个用空格隔开的正整数n, m，分别是城镇数目和道路数目；随后的m行对应m条道路，每行给出一对用空格隔开的正整数u, v，分别是该条道路直接相连的两个城镇的编号。简单起见，城镇从1到n编号。
> 当$n = 0$时输入结束。
> 注意：两个城市间可以有多条道路。
> $n \leq 1000$

> 输出格式:
> 对于每组数据输出一个整数表示还需要建设的道路条数。

样例输入	样例输出
4 2 1 3 4 3 3 3 1 2 1 3 2 3 5 2 1 2 3 5 999 0 0	 1 0 2 998

3. 修复公路

> A地区在地震过后，连接所有村庄的公路都造成了损坏而无法通车。政府派人修复这些公路。给出A地区的村庄数n，和公路数m，公路是双向的。并告诉你每条公路的连着哪两个村庄，并告诉你什么时候能修完这条公路。问最早什么时候任意两个村庄能够通车，即最早什么时候任意两条村庄都存在至少一条修复完成的道路（可以由多条公路连成一条道路）

> 输入格式:
> 第一行两个整数n, m
> 接下来m行，每行三个整数x, y, t，表示一条连接村庄x, y的道路在时间t修好。

> $x, y \leq n \leq 1000$
> $m, t \leq 10^5$

> 输出格式:
> 如果全部公路修复完毕仍然存在两个村庄无法通车，则输出-1，否则输出最早什么时候任意两个村庄能够通车。

样例输入	样例输出
1 2 6 1 3 4 1 4 5 4 2 3	5

四. 提高巩固

1. pSort

> 给定一个含有n个元素的数列，第i号元素开始时数值为i,位置i的元素可以与距离为$d[i]$的元素进行交换。再给定一个$1..n$的全排列，问初始的数列可否交换成给定的样式。

> 输入格式:
> 第一行一个整数n。
> 第二行n个互不相同的整数表示目标数列。
> 第三行n个整数表示$d[i]$。
> $n \leq 100$

> 输出格式:
> 若能交换到给定的样式则输出YES，否则输出NO。

样例输入	样例输出
7 4 2 5 1 3 7 6 4 6 6 1 6 6 1	YES

2. 搭配购买

> 明天就是母亲节了，电脑组的小朋友们在忙碌的课业之余挖空心思想着该送什么礼物来表达自己的心意呢？听说在某个网站上有卖云朵的，小朋友们决定一同前往去看看这种神奇的商品，这个店里有n朵云，云朵已经被老板编号为$1..n$，并且每朵云都有一个价值，但是商店的老板是个很奇怪的人，他会告诉你一些云朵要搭配起来买才卖，也就是说买一朵云则与这朵云有搭配的云都要买，电脑组的你觉得这礼物实在是太新奇了，但是你的钱是有限的，所以你肯定是想用现有的钱买到尽量多价值的云。

> 输入格式:
> 第一行三个整数n, m, w表示有n朵云，m种搭配，并现在有w元。
> 接下来n行，每行两个整数c_i, d_i表示第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u_i, v_i必须一起购买。
> $n, w \leq 10^4$
> $m \leq 5000$

> 输出格式:
> 一行一个整数表示可以获得的最大价值。

样例输入	样例输出
5 3 10 3 10 3 10 3 10 5 100 10 1 1 3 3 2 4 2	1

3. 信息传递

> 有n个同学（编号为1到n）正在玩一个信息传递的游戏。在游戏里每人都有一个固定的信息传递对象，其中，编号为i的同学的信息传递对象是编号为t_i
> 的同学。
> 游戏开始时，每人都只知道自己的生日。之后每一轮中，所有人会同时将自己当前所知的生日信息告诉各自的信息传递对象（注意：可能有人可以从若干人那里获取信息， 但是每人只会把信息告诉一个人，即自己的信息传递对象）。当有人从别人口中得知自己的生日时，游戏结束。请问该游戏一共可以进行几轮?

> 输入格式:
> 第一行包含一个整数n，表示有n个人。
> 第二行n个整数，第i个整数为t_i。
> $n \leq 2 * 10^5$

> 输出格式:
> 一行一个整数，表示游戏可以进行多少轮。

样例输入	样例输出
5 2 4 2 3 1	3