并查集

并查集是一种非常精巧而实用的树形数据结构，他主要是处理一些不相交集合的合并和查询的问题。不相交集合，顾名思义，就是两个集合的交集为空集的一些集合。比如1，3，5和2，4，6，他俩的交集为空集。就是不相交集合。像2，3，5和1，3，5就不是不相交集合。

使用并查集时，首先会存在一组不相交的动态集合 S={S1, S2,⋯,Sk}，一般都会使用一个整数表示集合中的一个元素。

每个集合可能包含一个或多个元素，并选出集合中的某个元素作为代表。每个集合中具体包含了哪些元素是不关心的，具体选择哪个元素作为代表一般也是不关心的。我们关心的是，对于给定的元素，可以很快的找到这个元素所在的集合（的代表），以及合并两个元素所在的集合，而且这些操作的时间复杂度都是常数级。

基本操作

所以说对于并查集，主要就是有如下的操作：

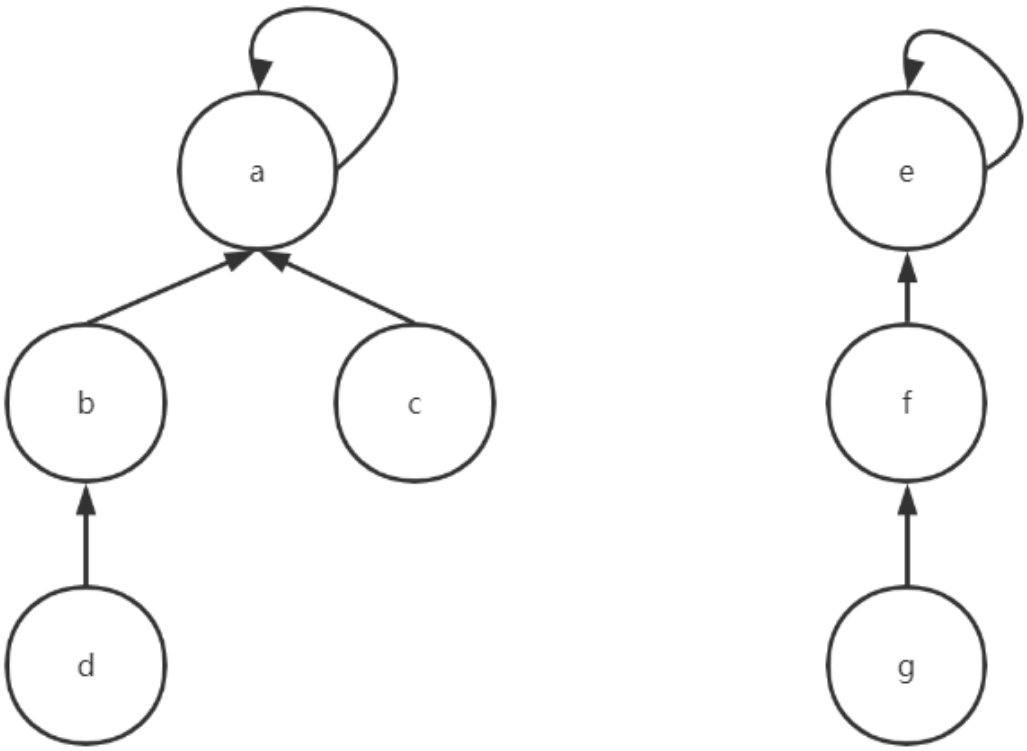
1、建立一个新的并查集，其中包含s个单元素集合

2、合并集合，把元素x和元素y所在的集合合并，要求x和y所在的集合不相交，如果相交就不合并。

3、找到元素x所在的集合的代表，这个操作也可以用于判断两个元素是否位于同一集合，只要将他们各自的代表比较一下就可以了。

树的表现形式

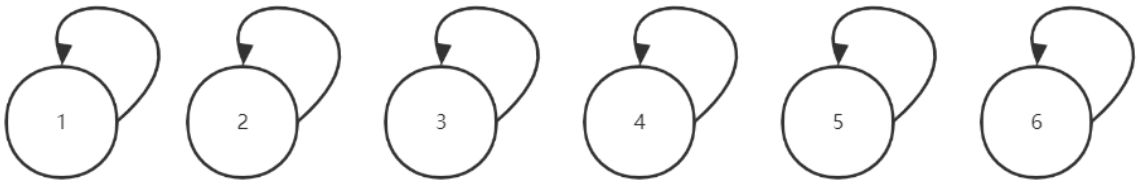
并查集的实现原理也比较简单，就是用树来表示一个集合，树的每个结点都是集合的一个元素，树根对应的那个结点就是该集合的代表。



比如上图中的两棵树，就分别对应两个集合，其中第一个集合为a,b,c,d,代表元素是a，第二个集合是e，f，g，代表结合是e。

树的节点表示集合中的元素，指针表示指向父节点的指针，根节点的指针指向自己，表示其没有父节点。沿着每个节点的父节点不断向上查找，最终就可以找到该树的根节点，即该集合的代表元素。

现在，假设使用一个足够长的数组来存储树节点，那么 makeSet 要做的就是构造出如下图的森林，其中每个元素都是一个单元素集合，即父节点是其自身：



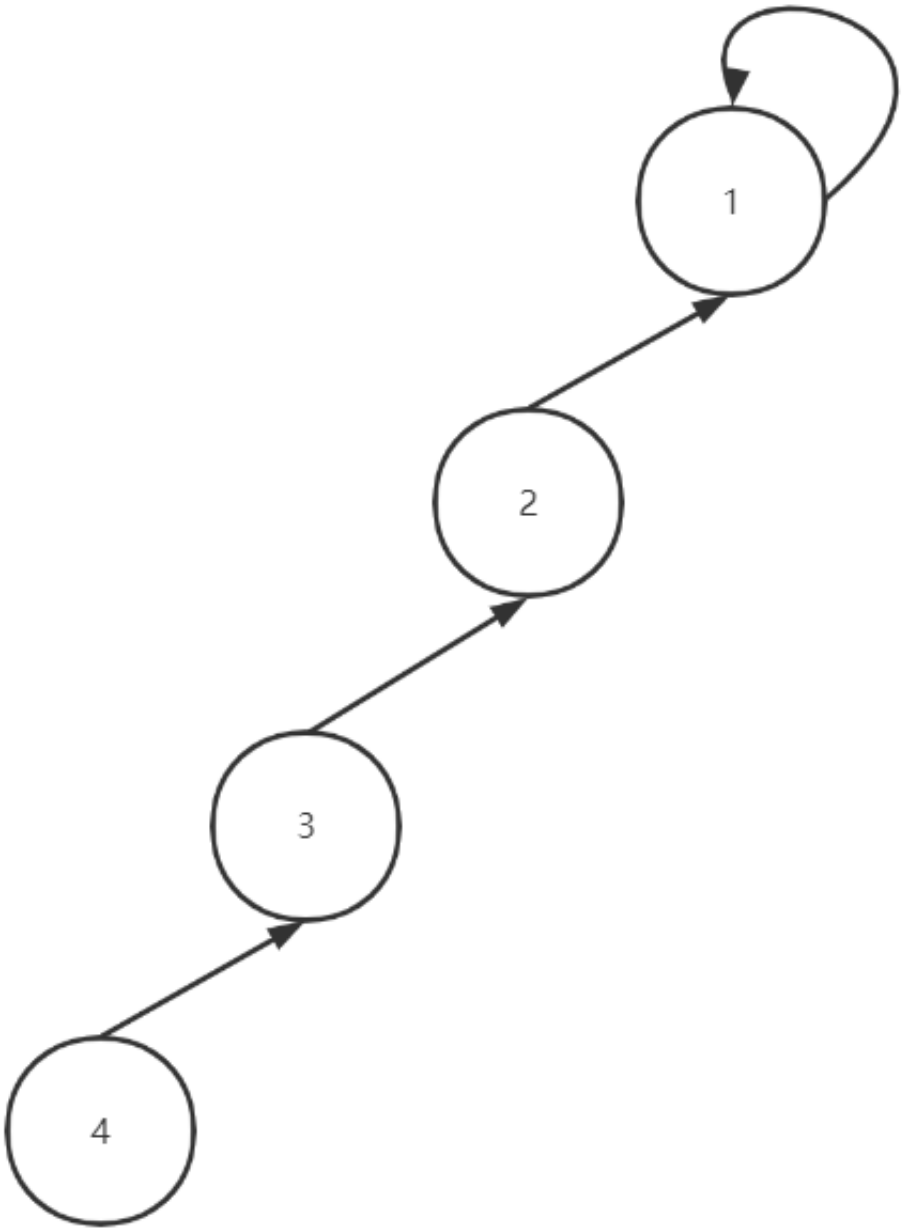
接下来就是find操作了

查找

查找操作就是找到i的祖先直接返回

合并

比如我们合并（4，3），（3，2），（2，1）。

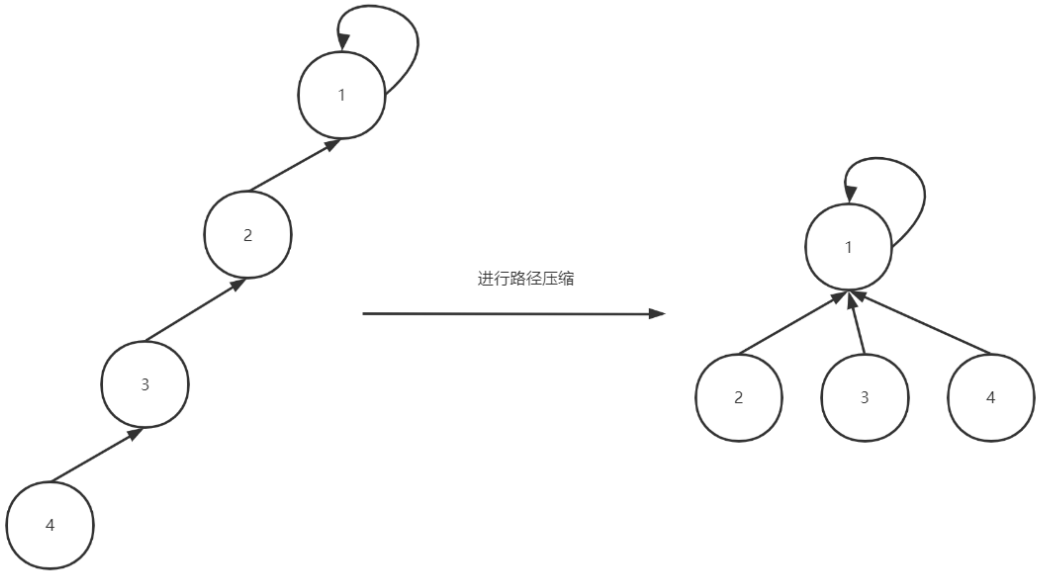


如果此时我们合并了一万个结点，这个时候，其实从时间复杂度来说，查找的次数就非常非常多了。

这个时候，我们就想出来了一个策略，路径压缩。

路径压缩

就是在每次查找时，令查找路径上的每个节点都直接指向根节点。


1
2
3
4
进行路径压缩
1
2
3
4