并查集

其实并查集顾名思义就是有`合并集合`和`查找集合中的元素`两种操作的关于数据结构的一种算法。它所处理的是`集合`之间的关系，即动态地维护和处理集合元素之间复杂的关系，当给出两个元素的一个无序对(a,b)时，需要快速`合并`a和b分别所在的集合，这其间需要反复"查找"某元素所在的集合。在这种算法中，n个不同的元素被分为若干组。每组是一个集合，这种集合叫做分离集合（disjoint set）。并查集支持查找一个元素所属的集合以及两个元素各自所属的集合的合并。

例如，有这样的问题：初始时n个元素分属不同的n个集合，通过不断的给出元素间的联系，要求实时的统计元素间的关系（是否存在直接或间接的联系）。这时就有了并查集的用武之地了。元素间是否有联系，只要判断两个元素是否属于同一个集合；而给出元素间的联系，建立这种联系，则只需合并两个元素各自所属的集合。这些操作都是并查集所提供的。

并查集支持操作

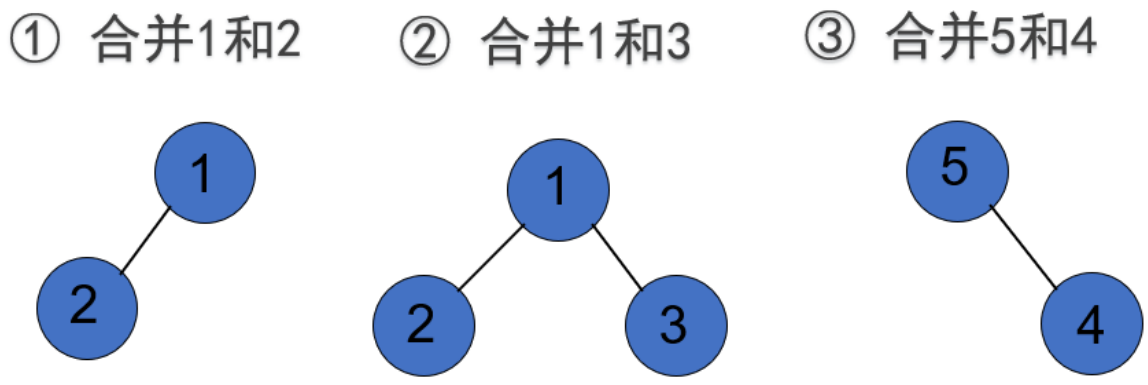
并查集的数据结构记录了一组分离的动态集合S={S1,S2,...,Sk}。每个集合通过一个代表加以识别，代表即该元素中的某个元素，哪一个成员被选做代表是无所谓的，重要的是：如果求某一动态集合的代表两次，且在两次请求间不修改集合，则两次得到的答案应该是相同的。

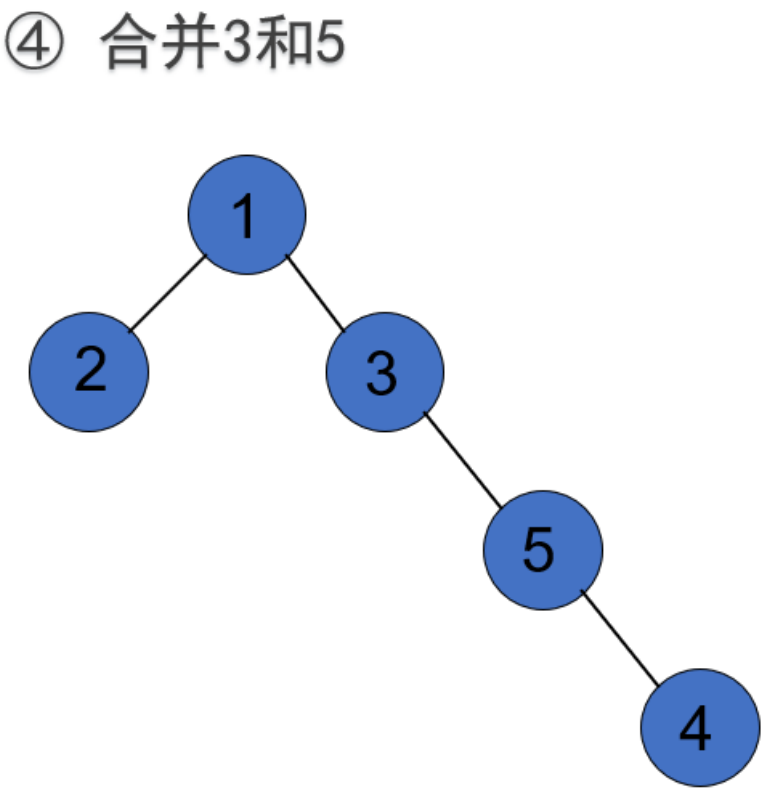
动态集合中的每一元素是由一个对象来表示的，设x表示一个对象，并查集的实现需要支持如下操作：

- buildset()：建立一个集合，每个集合内部只有一个元素，同时也是该集合的`代表`。
- merge(u,v)：将包含u和v的集合`合并为一个新的集合`
- find(x)：返回一个指向`包含x的集合的代表`

元素合并图解

用p[i]表示元素i的父亲结点，进行不断并到不同的集合中，初始p[i]=i。



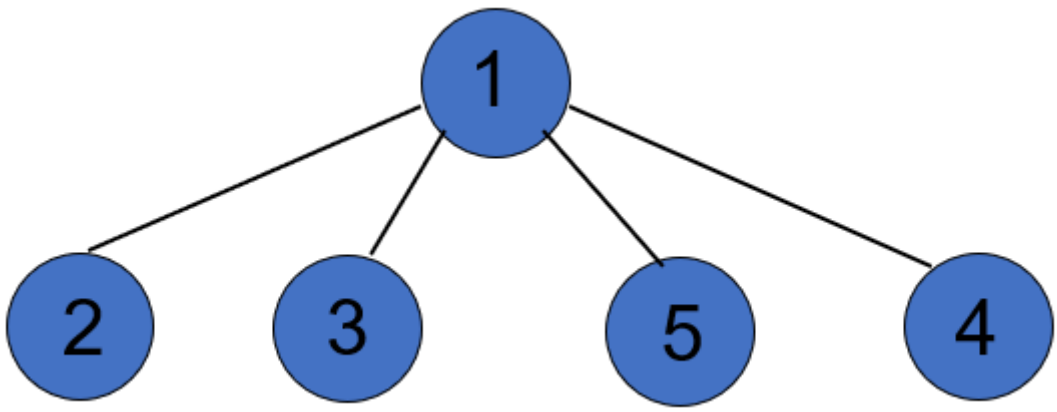


路径压缩

当数据比较特殊的时候，比如一条单链很长，数据这种`并`与`查`的方式可能会超时。

路径压缩就是将集合中元素指向一个代表，在找完根结点之后，在递归回来的时候顺便把路径上元素的父亲指针都指向根结点。

这就是说，我们在“合并5和3”的时候，不是简单地将5的父亲指向3，而是直接指向根节点1。



快乐刷题

- P432 家庭问题
- P455 格子游戏