# 并查集

* 并查集是一种树型的数据结构，用于处理一些不相交集合的合并问题。
* 并查集的主要操作有
* 1－合并两个不相交集合
* 2－判断两个元素是否属于同一个集合
* 3－路径压缩

练习

[Parity(ceoi99)](#ceoi99)

解析

Parity(ceoi99)

* 有一个01序列,长度<=1000000000,现在有n条信息,每条信息的形式是－a b even/odd。表示第a位到第b位元素之间的元素总和是偶数/奇数。
* 你的任务是对于这些给定的信息，输出第一个不正确的信息所在位置-1。信息的数目不超过5000。
* 如果信息全部正确，即可以找到一个满足要求的01序列，那么输出n。
* 从整个01序列肯定是无法入手的，因为它的长度高达109。
* 从范围比较小的n入手。也就是说我们需要对信息进行一些特殊的处理。
* a b even/odd，那么将元素b指向a-1，边的权值是even/odd。
* 下面我们由样例来说明一下这个处理方法。
* 建立sum数组，sum[i]表示从1到i之和是奇（true）还是偶（false），sum[0]=false。这样题目中给的任意问题（a,b）的答案都可以用sum[b] xor sum[a-1]表示。
* 开始我们并不知道sum[1..n]的值，不妨设为false，这时任意sum[a],sum[b]都是独立的。对于每对问答(a,b,c)，都可以知道sum[b] xor sum[a-1]=c,由此把sum[b]和sum[a-1]联系起来。这步操作可以用并查集完成，对于问答（a,b,c）如果sum[a-1],sum[b]不属于一个集合就把它们并起来，否则如果sum[a-1] xor sum[b]不等于c则说明出现矛盾，输出总句数，退出。
* 对于不出现矛盾的sum数组，对于每个集合分为两个部分，我们指定其中一个部分为true，另一个部分为false，则可以确定sum数组，利用sum[i] xor sum[i-1]可以求出第i位的数字，由于不同集合之间没有问答出现，所以此数列是一可行解，证明算法正确。