

# 序言

最近花了一点心思研究 **2-sat** 模型，看了很多论文博客等等，也在 POJ 上做了一点题。其实这个东西还挺好玩的，当然，前提是每道题你都有认真分析，认真想清楚模型的意义，搞明白为什么可以这样，而不是简单的知道怎样做，就套上一个模板了事，那样，是不是也太糟蹋这门科学了。

关于 **2-sat**，基本上所有人都会推荐两个资料：

[伍昱 由对称性解 2-sat 问题 \(ppt\)](#)

[赵爽 2-sat 解法浅析 \(pdf\)](#)

我当然也会推荐啦，好歹这也是引导我入门的东西，虽然说一开始看的并不是很懂，不过，还是要先膜拜一下两位大牛。

这篇文章主要是把我对 **2-sat** 的一些研究进行总结，我不太喜欢公式化的语言，所以，基本上会是白话陈述，主要包括以下五个内容：

- 1、概念的阐述
- 2、算法的阐述
- 3、算法的解释
- 4、构图的介绍
- 5、题目的分享

下面就进入正题。

# 概念

要知道 **2-sat** 是什么，我们先要知道什么是 **适定性 (Satisfiability)** 问题，适，就是合适，定，就是确定，适定性问题通俗的说就是确定是否可以满足所有的条件？或者说就是确定一个满足所有条件的方案。取英文的前三个字母，简称 **sat** 问题。

通俗的 **sat** 问题表述一般是这样的：有很多个集合，每个集合里面有若干元素，现给出一些取元素的规则，要你判断是否可行，可行则给出一个可行方案。如果所有集合中，元素个数最多的集合有  $k$  个，那么我们就说这是一个 **k-sat** 问题，同理，**2-sat** 问题就是  $k=2$  时的情况。

为什么研究 **2-sat** 呢？无可置疑的是它很有用。为什么不研究 **3-sat** 乃至  $k$  更大的情况呢，因为它们已经被证明为是 **NP 完全问题**了，在更多的情况下，**2-sat** 不仅仅是元素个数最多的集合含有 2 个元素，而是每个集合都含有 2 个元素，且这 2 个元素不允许同时取出，这篇文章主要讨论的就是这样一种特殊的模型。

# 算法

仍然这样，我的总结也从这道题开始。

题目大意：一国有  $n$  个党派，每个党派在议会中都有 2 个代表，现要组建和平委员会，要从每个党派在议会的代表中选出 1 人，一共  $n$  人组成和平委员会。已知有一些代表之间存在仇恨，也就是说他们不能同时被选为和平委员会的成员，现要你判断满足要求的和平委员会能否创立？如果能，请任意给出一种方案。（POI 0106）

正如你看到的那样，题目给的非常裸，其实，**2-sat** 的题目都没有隐藏的那么深，不像网络流那样很难发现。

对于这个问题，暴力当然可以解决了，其实说的更好听一点，也就是在图上进行暴力，噢，是不是感觉很高级的样子，嗯，这其实就是一个很容易理解的算法了。

## 算法一：持之以恒

假设 1A、1B 为 1 号党的两名党员，2A、2B（噢，不准笑）为 2 号党的两名党员，且 1A 与 2A 之间存在仇恨！那么，这意味着什么？如果选择 1A，就只能选择 2B 了，如果选择了 2A 呢？那同理就只能选择 1B 了，那么，我们可以这样构建一个图，在 1A 到 2B 之间搭上一条边，2A 到 1B 之间搭上一条边，这条边的含义就是：必须。

也就是说，沿着一条路径，如果一个点被选择了，那么这条路径以后的点都将被选择，那么，出现不可行的情况就是，存在一个党派两个党员都被选择了，那么，我们只需要枚举一下就可以了，数据不大，答案总是可以出来的。

这么一个简单的算法，放在第一个说，看来你已经知道它不是重点了，但是，若题目要求的是字典序最小的方案的话，那就只能选择这个算法了。

## 算法二：没有名字

构图还是跟第一个算法一样，但是在解图的过程中进行了非常漂亮的优化。

为了说清楚传递，我们先来回忆一下命题的概念：命题、逆命题、否命题、逆否命题。

概念就不说了，举了例子也好：

**命题：**三角形内角和为  $180^\circ$

**逆命题：**内角和为  $180^\circ$  的是三角形

**否命题：**不是三角形的内角和不为  $180^\circ$

**逆否命题：**内角和不是  $180^\circ$  的不是三角形

有个很显然的结论就是，命题与逆否命题的真假性是一样的（又有人要说我啰嗦了噯）

也许，你正在很奇怪我为什么要扯到这个东西，虽然我看的所有资料都没有介绍过，也没人把这两东西扯到一起，不过，我还就真的觉得它们就是一路的，怎么说？



选 A 就必须选 B，那么我们先“逆一下”，从 B 到 A，再“否一下”，不选 B 就一定不选 A，这是显然成立的，也就是说，在构好的图中，“必须”是沿着边正方向传递下去的，“禁止”是沿着边反方向传递下去的，那么每次我们选择了一个点，就把它的对立点（即同集合的另一元素）标记为不可选，然后呢，我们把选择标记正向传递，再把禁止标记逆向传递，这是符合题意的高效做法，但是……还是很麻烦，竟然要双向传递，而且……你怎么知道一开始要选择哪一个节点才好呢？囧。

不要急，请先达成一点共识：

- 1、如果很多节点已经处在一个强连通分量内了，也就是它们两两均可互达了，那么意味着选择了其中的一个，整个强连通分量都将被选择，如果不选择其中的一个，那么整个强连通分量内都不能够被选择。
- 2、如果一个集合的两个元素处在了同一个强连通分量内，也就是说要么都不选，要么都被选，那么题目要求的选出一个就一定无法成立了。
- 3、题中不管给出哪两个不能同时选，都不能改变我们连的边都是对称的这一事实，只要 A 到 B 有边，那么 B 的对立节点到 A 的对立节点就有边。
- 4、边的对称意味着图的对称，原图的对称意味着哪怕用强连通分量缩点后的图一样对称，所以对立节点的概念也可以应用在强连通分量上。

5、为了叙述方便，以下简称强连通分量为块，注意：**真正的块指的是点双连通分量**，这里只是一种形象化的表述，是不规则的！

算法的过程如下：

构图

更具体的后面再说

缩点

Tarjan 算法缩点，将所有的边反过来（为什么？后面有噢）

判可行

枚举集合的两个元素，看其是否处于不同的块内，若否的话则给出不可行信息

记录矛盾

这里所说的矛盾不是题中给出的两个人之间有仇恨，那样的边是实际存在，我们这里说的矛盾是指若两个块分别含有两个对立节点，也就是说一个集合的两个元素分布在了两个不同的块中，那么这两个块就是矛盾的，即不可能同时被选择，这样一种矛盾关系是不存在于边中的，是不依赖于输入的数据的，我们要找到与一个块对立的块，并把它们保存下来。

拓扑排序

将缩点后的图进行拓扑排序（排的是块而不是节点）

构造方案

按照拓扑序列的顺序，依次访问所有块，若一个块未被标记，将其标记为“选择”，不传递“选择”标记，将被选块的对立块标记为“不选择”，将其“不选择”标记沿着边正向传递。（不是逆着边么？哼，图已经被反过来了，你到底有没有认真看呐！囧）

没了？没了，真没了。我知道你很想问为什么，= =，ps：是不是觉得我特别啰嗦……

唉，啰嗦惯了，看到那些“一点都不啰嗦的论文我就想\*\*”。回到正题，我们继续。

# 解释

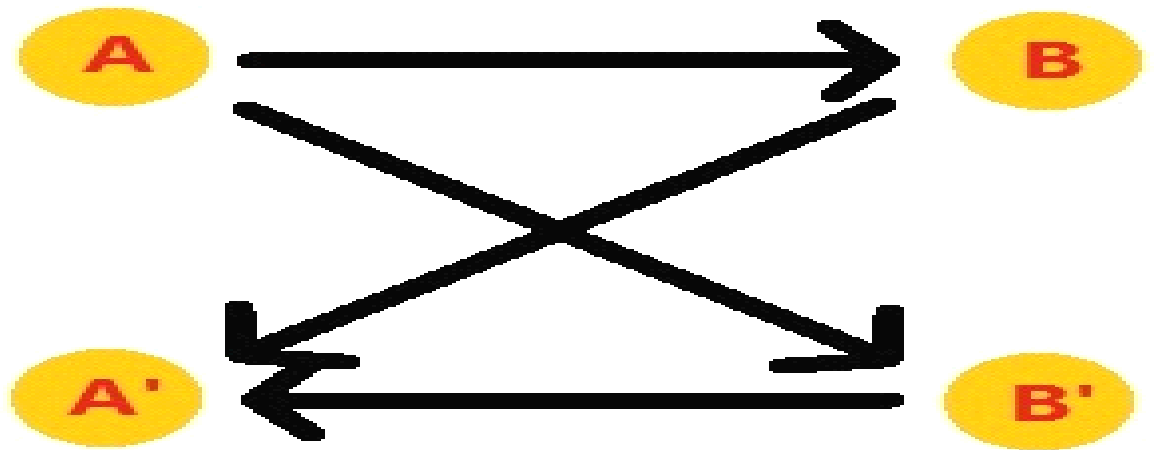
## 1、为什么要用反边存缩点后的图？

-- 考虑到算法传递标记的时候，“选择”标记是没有进行传递的，也就是说正向边是没有利用到的，传递的都是“不选择”标记，也就是说走的都是反边，但是，邻接表存边，怎么走得反边呢？那么既然正向边没有用，就把整个图反过来存就好了，并且，这个反图还真是反得好啊！为什么呢，往下看。

## 2、为什么要拓扑排序呢？

-- 不该来的还是来了，这可真是令人恶心的问题，哪怕到我现在写，我也无法对它有种深深的认同，你懂的，有些东西，哪怕你并说不出它为什么如此，但是你能感受到，能够认同到，能够抓住它的走向，深信其的正确，但是，有的东西，哪怕你已经能够说出为什么了，还是无法对其有深深的认同感，总觉得有种别扭、不舒服的感觉在，少废话，继续。

-- 观察下面的这幅图（注意：此图为原图，非缩点后存的反图）



如果我们没有按照拓扑序列来选择块的话：访问到 A，发现其无标记，于是把它标记为可选择，然后把 A' 标记为不可选择，然后呢，进行标记传递，但是问题出现了，若进行“选择”标记传递，则 B 和 B' 将被标记为选择，若进行“不选择”标记传递，则 B 和 B' 将被标记为不选择，这又选择又不选择的，不就矛盾了吗？难道是因为此图无解吗？但是根据我们判断无解的标准，即两对立节点处在同一块中，此图是有解的，不存在节点可以互达，究其原因，是因为我们的构造方法不对，A 处在了整个边的上游地带，先就将其确定为选择对原图的影响太大了，根本就无法保证接下来还能进行正确的抉择。所以，正确的方法应该是按照拓扑顺序，从影响小的开始抉择，在转成反边后，A' 会处在拓扑序列的首位，将 A' 标记为选择，将 A 标记为不选择，都不能传递，再将 B 标记为选择，B' 标记为不选择，当然，将 B 标记为不选择，B' 标记为选择也是可以的，这完全看拓扑排序的编写规则，毕竟我们是访问到一个无标记的就将其标记为选择的，那么谁先出现谁就被选择了，自然界的规律哈哈。

### 3、为什么只传递“不选择”标记，不传递“选择”标记？

-- 还记得我们的选择规则吗？好像已经重复了好多遍了……找到一个未被标记的，就将其标记为选择，然后将对立块标记为不选择，再把不选择标记传递，如果我们选择了 B，那么它的后代节点 A'（此处为原图的后代，即上图所示）一定被选择了的？为什么呢？如果它的后代节点 A' 是没有被选择的，那么它的后代节点的对立节点 A 肯定是被选择了的，根据边的对称性，A' 是 B 的后代节点，那么 B' 就是 A 的后代节点，既然 A 被选择了，那么 B' 也要被选择，那么根据顺序，B 会被标记为不可选择，那么我们在访问到 B 的时候，就不会把它认定为未标记，那也就不存在将其标记为选择了，这与假设是矛盾的，显然不可行。所以，既然我们选择一个节点的时候，它的后代都已经被选择好了，那么我们就需要进行“选择”标记的传递了，况且，到标记传递的时候，我们都已经存的是反图了，就算你想传递也没办法了……囧。总的来说，就是虽然没有传递选择标记，但是这种传递的性质是满足的。

### 4、标记法构造方案的时候为什么不会同时选定一个集合的两个元素？

-- 嗯，我想这跟上一个问题是一样的，如果你还心存疑惑，Please read it again.

### 5、标记法构造方案的时候为什么不会把同一集合两个元素都标记为不选择？

-- 我们不妨把选择一个块，然后将其对立块标记为不选择这一行为叫做直接标记，把其对立块进行“不选择”标记传递时标记块叫做间接标记，下面就容易说话了.....

**（1）若两对立块是在某块被标记为选择后，同时被其直接标记为不选择的：**

那么我们想想它们为什么会被其直接标记？当然是因为它们的内部有与标记为选择的块对立的节点，在记录矛盾这一步骤中被记录了下来，所以才会这样的，但是再定睛一看，凭什么被标记为选择的块中某两节点的对立节点分布在了不同的块中？还记得我们达成的共识吗？边的对称意味着块的对称，那么这两个被直接标记为不选择的块本就不应该分开，这种情况是不成立的。**也就是说，这两个被标记为“不可选”的块都不是直接标记的。**

这同时还说明了很重要的一点，与一个块直接矛盾的块存在且仅存在一个，所以我们记录矛盾块用的不是邻接表，而只是一个数组就可以了。

**（2）若两对立块是在某块被标记为选择后，其中一个块被其直接标记，另一块被其间接标记：**

因为其中一块被其直接标记，故其间存在一对对立节点，另一块与被直接标记的这一块为对立块（根据大前提），那么又出现了存在于被直接标记的块中的两个节点，它们的对立节点分散在了两个块中，这又矛盾了。**也就是说，这两个被标记为“不选择”的块都不是被直接标记的。**

**（3）若两对立块是在某块被标记为选择后，同时被其间接标记为不可选择的：**

这么说来，被标记为选择的块的对立块（假设为 O），它到这两个块（假设为 A，A'）都是有路径的，因为要进行不选择标记传递啊（此处为反图叙述），但是根据图的对称性，既然 O 到 A 有路径，那么 A' 就到 O，然而，O 到 A' 也有路径，那么两个块就显然是一个块，这是与假设相悖的。**也就是说，这两个被标记为“不可选”的块也都不是间接标记的。**

那么，总的来说，在将一个块标记为“选择”之后，在直接标记与间接标记中都不会出现上述情况。那在不同的块被标记为“选择”之后，会出现上述情况吗？

**（4）在整个操作中会把两个对立块都标记为不选择吗？**

至此，我们已经可以确定的就是，在一次选择与传递过程中，是不会出现这样的情况，若是放在整个的构造过程中，我们也可以确信的，两个对立块的任何一个都不会被别的块直接标记为不可选，如果一个块已经被标记为不可选了，那么只能是它的对立块被标记为可选了，才将其直接标记，同时，那些可能再度将其标记为不可选的块，会在它的间接标记下，死亡 ..... 嗯，也就是被夺去标记别人的资格，为什么？因为图是对称的！

这个问题就这样愉快的解决了！

## 6、都说用对称性解决 2-sat 问题，到底哪里用到了对称性？

-- 是的，这个问题真的是太蠢的，不知道你是不是有跟我一样的想法，至少我就纠结过好久，唉，让我再感叹一次吧！愚蠢的问题啊！算法虽然没有直接用到对称性来实现，但是对称性是算法的基石啊！如果原图不是对称的！那前面的证明就都不成立，那么算法正确性就无法保证了！真是太 2 的问题了，谁说一定要在算法过程中体现才叫用到啊，唉，真是 2 死了。

## 模型

当然，我还是很希望你看到这里已经懂了，不然，肯定是我叙述的太没水准了。== 如果不懂的话，还是弄懂再往下走为妙，毕竟，（词乏了，略去）

如果你以为 2-sat 就只是和平委员会这样的题目的话，那你就错了，但是，你以为它比和平委员会多很多的话，那你也错了，其实，常用的模型没有太多。

## 模型一：两者（A，B）不能同时取

那么选择了 A 就只能选择 B'，选择了 B 就只能选择 A'

连边  $A \rightarrow B'$ ， $B \rightarrow A'$

## 模型二：两者（A，B）不能同时不取

那么选择了 A'就只能选择 B，选择了 B'就只能选择 A

连边  $A' \rightarrow B$ ， $B' \rightarrow A$

## 模型三：两者（A，B）要么都取，要么都不取

那么选择了 A，就只能选择 B，选择了 B 就只能选择 A，选择了 A'就只能选择 B'，选择了 B'就只能选择 A'

连边  $A \rightarrow B$ ， $B \rightarrow A$ ， $A' \rightarrow B'$ ， $B' \rightarrow A'$

## 模型四：两者（A，A'）必取 A

那么，那么，该怎么说呢？先说连边吧。

连边  $A' \rightarrow A$

你想出来为什么了吗？也许你在想，哇塞，这样就保证了在反图中 A 在拓扑序中靠前，然后就会先选择 A，呵呵，你错了。

虽然，我在一些人的博客中见到了这样的解释，但是，我还是非常负责的告诉你，这是不全面的。

我们从 A'向 A 连边，要的不仅是拓扑序，还有判可行与否。在 2-sat 图当中，若该图是可行的，就意味着如果从 A 到 A'有路径的话，A 到 A 是没有路径的，因为如果有路径，它们就成一个块了，不会被判为可行，那么，如果 A 到 A'是有路径的话，在反图中，A'就到 A 有路径，那么拓扑里，A'就会因为靠前被标记为“选择”，并未满足条件。并且，我们应当确信的是，解的多情况依赖的是拓扑排序的多情况，不按拓扑序来的话，解就是错误的，或者说不成立的，也就是说，我们拓扑序先选择 A 的话，就会导致别的约束条件的不成立，那么在我们引入了 A'到 A 的这条边后，A 就与 A'在同一块中了，算法会报告不可行信息。若是原图本来就满足 A 到 A'有路径，然而 A'到 A 无路径的话，我们添一条边，只是确定了其有解，但丝毫不会影响到拓扑排序，只有当 A 到 A'之间根本不存在路径的时候，才会影响到其拓扑排序，所以，真相是这样的。

是不是有人已经开始疑惑，讲了这么久的对称性，整个算法都是依赖对称性才得以成立的，那么怎么一下引入一条边让图不对称了算法还成立呢！关于这个问题，其实很好解释，仔细想想，对称性确保的是什么？它确保的可是原图有解，则一定可以构造出来。现在我们引入了一条边  $A' \rightarrow A$ ，若通过上述判断，让其无解了，那么对后面显然是没有影响的，毕竟算法都没执行下去了，还算什么影响。如果还有解呢？这说明了什么？这说明了 A'到 A 原本就存在一条路径，或者 A'到 A 之间根本就没有路径。如果有路径的话，那么我多增加一条有区别吗？它会影响算法的任何一步吗？显然不会啊，那如果原本没有路径的话呢？没有路径意味着谁在拓扑序列中靠前都是可能的，在有了该边后（反边为  $A \rightarrow A'$ ），A 将在拓扑序列中靠前，被标记为“选择”，那么我们会对 A'进行直接标记，此时，A'到 A 是没有路可走的，它根本就无法访问到 A，所以在标记的这个过程中，这条路径根本就没有影响到图的任何对称性，在拓扑排序后，你完全可以当其不存在。

模型其实都大同小异，还是看自己去变幻，比如模型一和模型二其实就是一样的，只是针对的点不同而已。一般，如果 2-sat 感觉很明显的话，就用 2-sat 做好了，不过，千万记得考虑好如何降低构图的复杂度噢！复杂度如果太高，就还是再另辟蹊径吧，也许有更加好的方法噢。

## 题目

都源自 POJ，大家可以自行练习，如果要对拍什么的，这里也有程序，题目按从简单到简单的顺序排序（噢 .....其实是推荐做题顺序），大家，一起好好加油。

[POJ 3207](#)

[POJ 3683](#)

[POJ 3678](#)

[POJ 3648](#)

[POJ 2723](#)

[POJ 2749](#)

完、

<http://blog.csdn.net/jarjingx/article/details/8521690>