2-SAT

Definition：给定 n 个变量 $a_i\in\{0,1\}$，同时给出若干条件：$(\text{not})a_i \text{ opt } (\text{not})a_j$，其中 $\text{opt}\in\{\text{AND},\text{OR},\text{XOR}\}$；求解 2-SAT 问题就是求出一组合法的 $\{a_i\}$.

Idea：对于第 i 个变量，用两个点 $2i$ 和 $2i+1$ 分别表示 $a_i=0$ 和 $a_i=1$；求解一组合法的 $\{a_i\}$ 就是对每个 a_i，要么选择 $2i$，表示 $a_i=0$，要么选择 $2i+1$，表示 $a_i=1$；用有向边 $x\to y$ 表示选择了 x 就必须选择 y.

考虑把给定条件转化为连边：

条件	连边
$a_i=0$	$2i+1\to 2i$
$a_i=1$	$2i\to 2i+1$
$a_i \text{ XOR } a_j=1$ $a_i\neq a_j$	$2i+1\to 2j$ $2j+1\to 2i$ $2i\to 2j+1$ $2j\to 2i+1$
$a_i \text{ XOR } a_j=0$ $a_i=a_j$	$2i+1\to 2j+1$ $2j+1\to 2i+1$ $2i\to 2j$ $2j\to 2i$
$a_i \text{ OR } a_j=1$ $a_i=1 \text{ or } a_j=1$ a_i,a_j 至少一个是 1	$2i\to 2j+1$ $2j\to 2i+1$
$a_i \text{ OR } a_j=0$ $a_i=a_j=0$	$2i+1\to 2i$ $2j+1\to 2j$
$a_i \text{ AND } a_j=0$ $a_i=0 \text{ or } a_j=0$ a_i,a_j 至少一个是 0	$2i+1\to 2j$ $2j+1\to 2i$
$a_i \text{ AND } a_j=1$ $a_i=a_j=1$	$2i\to 2i+1$ $2j\to 2j+1$
$a_i=1 \text{ or } a_j=0$	$2i\to 2j$ $2j+1\to 2i+1$
$a_i=0 \text{ or } a_j=1$	$2i+1\to 2j+1$ $2j\to 2i$

这样我们得到了一个有向图。这个有向图中，属于同一个强连通分量的点要么同时被选，要么同时不被选。因此，如果存在某个 i，$2i$ 和 $2i+1$ 都在同一个强连通分量中，那么问题无解。使用 **Tarjan** 算法求强连通分量。

否则，我们得到一个 **DAG**，并且根据 **Tarjan** 算法的特性，我们新图的标号正好是**反向的拓扑序**。此时，若 x,y 处于同一个连通分量中，且 x 的拓扑序小于 y 的拓扑序，那么选择了 x，就一定要选择 y. 考虑情况：i 对应的两个点 $2i,2i+1$ 在同一个连通分量中，为了不发生矛盾，我们只能选择拓扑序更大者，即 **Tarjan** 标号更小者。可以证明，对每个 i 这么选之后，一定得到一个可行解。

Complexity：$O(V+E)$

ATT：开 2 倍空间。

Code：

```
int main(){
    // ... input & build edges
    for(int i = 2; i <= (n<<1|1); i++)
        if(!dfn[i]) tarjan(i);
    for(int i = 1; i <= n; i++){
```

```
        if(belong[i<<1] == belong[i<<1|1]){
            puts("IMPOSSIBLE");
            return 0;
        }
    }
    puts("POSSIBLE");
    for(int i = 1; i <= n; i++)
        printf("%d ", belong[i<<1] < belong[i<<1|1] ? 0 : 1); // the ith variable is 0/1
    // ...
}
```