Bouton's Theorem

结论：对于一个 Nim 游戏的局面(a1,a2,...,an)，它是 P-position 当且仅当 a1^a2^...^an=0，其中^表示异或(xor)运算。

若 a1^a2^...^an!=0 则令 a1^a2^...^an=k， k 的最高位一定来自于某个 ai 的二进制，这时有 ai^k<ai， 如果我们把 ai 这堆石子拿掉(ai-ai^k)个石头变成 ai^k， 则此时 a1^a2^...^an^k=k^k=0， 故 a1^a2^...^an!=0 时可以保证一步走到 P-position， 所以此时状态为 N-positon

https://www.luogu.com.cn/problem/P2197

```
#include <cstdio>

using namespace std;

int n, ans;

void solve(){
    scanf("%d",&n);
    ans = 0;
    for(int i = 1, a ; i <= n ; ++i){
        scanf("%d",&a);
        ans ^= a;
    }
    if(!ans) puts("No");
    else puts("Yes");
}

int main(){
    int t;
    scanf("%d",&t);
    while(t--) solve();
    return 0;
}
```

SG 函数

大部分公平组合游戏可以转换为有向图游戏，在一个有向无环图中，双方轮流推动棋子从起点开始到下一个点，不能走的一方就输掉游戏。

定义在这样的 DAG 中，一个点x的k个后继为$y1, y2, \cdots, yk$

则 SG(x)=mex{SG(y1),SG(y2),…,SG(yk)}

SG 定理

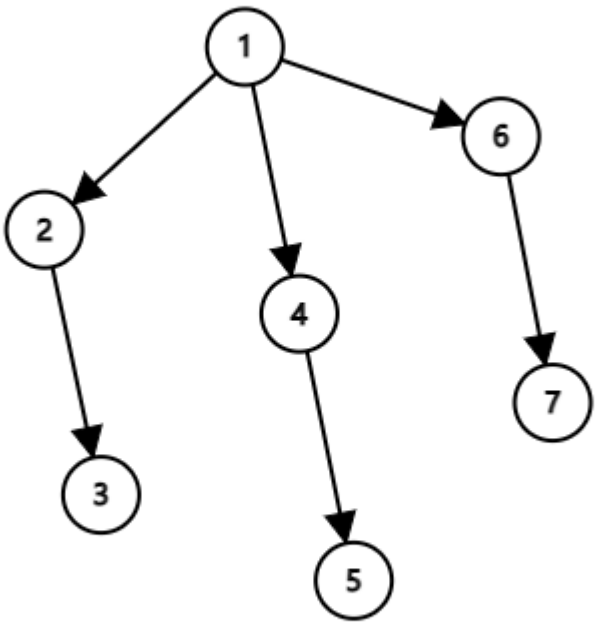
设一个博弈游戏的起点在 DAG 中的起点为 s1,s2,…,sk，则有定理：当 x=SG(s1)^SG(s2)^…^SG(sk)≠0 时，这个游戏是先手必胜的，同时把 x 值成为这个组合游戏的 SG 值。

Bouton's Theorem 可以理解成有 n 个游戏 xi，分别为在一堆 ai 个石子的石堆中取任意颗石子，不能取的一方判输，显然 SG(xi)=ai，把 n 个游戏的 SG 值异或就可以由 SG 定理得到 Bouton's Theorem

树上删边游戏

① 对于一条有 n 个节点的链来说，可以删掉 1~n-1 个节点，可以比作 nim 游戏，它的 SG 值为 n-1，如果增加一个节点，它的 SG 值就会加 1

② 由父亲走向的下一个节点可以表示为一个独立的游戏，所以父亲的 SG 值等于异或它的每一个子节点的 SG 值+1



③ 根据 SG 定理，sg1=0 的时候为 P-position,sg1≠0 的时候为 N-position

D - Game on Tree (atcoder.jp)

```
#include <cstdio>

using namespace std;

const int MAXN = 1e5 + 5;

int head[MAXN], tot;

struct Edge{
```

```
    int to, next;
}G[MAXN<<1];

inline void addEdge(int u, int v){
    G[++tot].to = v;
    G[tot].next = head[u];
    head[u] = tot;
}

int n, sg[MAXN];

void DFS(int np, int fat){
    for(int i = head[np], to ; i ; i = G[i].next){
        to = G[i].to;
        if(to==fat) continue;
        DFS(to,np);
        sg[np] ^= sg[to]+1;
    }
}

int main(){
    scanf("%d",&n);
    for(int i = 1, u, v ; i < n ; ++i){
        scanf("%d %d",&u,&v);
        addEdge(u,v);
        addEdge(v,u);
    }
    DFS(1,0);
    if(sg[1]) puts("Alice");
    else puts("Bob");
    for(int i = 1 ; i <= n ; ++i) printf("%d\n",sg[i]);
    return 0;
}
```