博弈论

巴什博奕

> 有 N 个石子，两名玩家轮流行动，按以下规则取石子：
>
> 规定：每人每次可以取走 $X(1 \le X \le M)$ 个石子，拿到最后一颗石子的一方获胜。
>
> 双方均采用最优策略，询问谁会获胜。

> 两名玩家轮流报数。
>
> 规定：第一个报数的人可以报 $X(1 \le X \le M)$ ，后报数的人需要比前者所报数大 $Y(1 \le Y \le M)$ ，率先报到 N 的人获胜。
>
> 双方均采用最优策略，询问谁会获胜。

- $N = K \cdot (M+1)$ （其中 $K \in \mathbb{N}^+$ ），后手必胜（后手可以控制每一回合结束时双方恰好取走 $M+1$ 个，重复 K 轮后即胜利）；
- $N = K \cdot (M+1) + R$ （其中 $K \in \mathbb{N}^+, 0 < R < M+1$ ），先手必胜（先手先取走 R 个，之后控制每一回合结束时双方恰好取走 $M+1$ 个，重复 K 轮后即胜利）。

扩展巴什博弈

> 有 N 颗石子，两名玩家轮流行动，按以下规则取石子:。
>
> 规定：每人每次可以取走 $X(a \le X \le b)$ 个石子，如果最后剩余物品的数量小于 a 个，则不能再取，拿到最后一颗石子的一方获胜。
>
> 双方均采用最优策略，询问谁会获胜。

- $N = K \cdot (a+b)$ 时，后手必胜；
- $N = K \cdot (a+b) + R_1$ （其中 $K \in \mathbb{N}^+, 0 < R_1 < a$ ） 时，后手必胜（这些数量不够再取一次，先手无法逆转局面）；
- $N = K \cdot (a+b) + R_2$ （其中 $K \in \mathbb{N}^+, a \le R_2 \le b$ ） 时，先手必胜；
- $N = K \cdot (a+b) + R_3$ （其中 $K \in \mathbb{N}^+, b < R_3 < a+b$ ） 时，先手必胜（这些数量不够再取一次，后手无法逆转局面）；

Nim 博弈

> 有 N 堆石子，给出每一堆的石子数量，两名玩家轮流行动，按以下规则取石子：
>
> 规定：每人每次任选一堆，取走正整数颗石子，拿到最后一颗石子的一方获胜（注：几个特点是**不能跨堆**、**不能不拿**）。
>
> 双方均采用最优策略，询问谁会获胜。

记初始时各堆石子的数量 $(A_1, A_2, \dots, A_n)$ ，定义尼姆和 $Sum_N = A_1 \oplus A_2 \oplus \dots \oplus A_n$ 。

当 $Sum_N = 0$ 时先手必败，反之先手必胜。

Nim 游戏具体取法

先计算出尼姆和，再对每一堆石子计算 $A_i \oplus Sum_N$ ，记为 X_i 。

若得到的值 $X_i < A_i$ ，X_i 即为一个可行解，即**剩下 X_i 颗石头，取走 $A_i - X_i$ 颗石头**（这里取小于号是因为至少要取走 1 颗石子）。

Moore's Nim 游戏（Nim-K 游戏）

> 有 N 堆石子，给出每一堆的石子数量，两名玩家轮流行动，按以下规则取石子：
>
> 规定：每人每次任选不超过 K 堆，对每堆都取走不同的正整数颗石子，拿到最后一颗石子的一方获胜。
>
> 双方均采用最优策略，询问谁会获胜。

把每一堆石子的石子数用二进制表示，定义 One_i 为二进制第 i 位上 1 的个数。

以下局面先手必胜：

对于每一位， $One_1, One_2, \dots, One_N$ 均不为 $K+1$ 的倍数。

Anti-Nim 游戏（反 Nim 游戏）

> 有 N 堆石子，给出每一堆的石子数量，两名玩家轮流行动，按以下规则取石子：
>
> 规定：每人每次任选一堆，取走正整数颗石子，拿到最后一颗石子的一方**出局**。
>
> 双方均采用最优策略，询问谁会获胜。

- 所有堆的石头数量均不超过 1 ，且 $\boldsymbol{Sum_N = 0}$ （也可看作"且有偶数堆"）；
- 至少有一堆的石头数量大于 1 ，且 $\boldsymbol{Sum_N \neq 0}$ 。

阶梯 - Nim 博弈

> 有 N 级台阶，每一级台阶上均有一定数量的石子，给出每一级石子的数量，两名玩家轮流行动，按以下规则操作石子：
>
> 规定：每人每次任选一级台阶，拿走正整数颗石子放到下一级台阶中，已经拿到地面上的石子不能再拿，拿到最后一颗石子的一方获胜。
>
> 双方均采用最优策略，询问谁会获胜。

对奇数台阶做传统 `Nim` 博弈，当 $\boldsymbol{Sum_N = 0}$** 时先手必败，反之先手必胜。**

SG 游戏（有向图游戏）

我们使用以下几条规则来定义暴力求解的过程：

- 使用数字来表示输赢情况，0 代表局面必败，非 0 代表**存在必胜可能**，我们称这个数字为这个局面的SG值；
- 找到最终态，根据题意人为定义最终态的输赢情况；
- 对于非最终态的某个节点，其SG值为所有子节点的SG值取 `mex` ；
- 单个游戏的输赢态即对应根节点的SG值是否为 0 ，为 0 代表先手必败，非 0 代表先手必胜；
- 多个游戏的总SG值为单个游戏SG值的异或和。

使用哈希表，以 $\mathcal{O}(N+M)$ 的复杂度计算。

```cpp
int n, m, a[N], num[N];
int sg(int x) {
    if (num[x] != -1) return num[x];

    unordered_set<int> S;
    for (int i = 1; i <= m; ++ i)
        if(x >= a[i])
            S.insert(sg(x - a[i]));

    for (int i = 0; ; ++ i)
        if (S.count(i) == 0)
            return num[x] = i;
}
void Solve() {
    cin >> m;
    for (int i = 1; i <= m; ++ i) cin >> a[i];
    cin >> n;

    int ans = 0; memset(num, -1, sizeof num);
    for (int i = 1; i <= n; ++ i) {
        int x; cin >> x;
        ans ^= sg(x);
    }

    if (ans == 0) no;
    else yes;
}
```

Anti-SG 游戏（反 SG 游戏）

SG 游戏中最先不能行动的一方获胜。

以下局面先手必胜：

- **单局游戏的SG值均不超过 1 ，且总SG值为 0；**
- **至少有一局单局游戏的SG值大于 1 ，且总SG值不为 0 。**

在本质上，这与 Anti-Nim 游戏的结论一致。

Lasker’s-Nim 游戏（ Multi-SG 游戏）

> 有 N 堆石子，给出每一堆的石子数量，两名玩家轮流行动，每人每次任选以下规定的一种操作石子：
>
> - 任选一堆，取走正整数颗石子；
> - 任选数量大于 2 的一堆，分成两堆非空石子。
>
> 拿到最后一颗石子的一方获胜。双方均采用最优策略，询问谁会获胜。

本题使用SG函数求解，SG值定义为：

$$SG(x)=\begin{cases}x-1 & ,x \mod 4=0\\ x & ,x \mod 4=1\\ x & ,x \mod 4=2\\ x+1 & ,x \mod 4=3\end{cases}$$

Every-SG 游戏

> 给出一个有向无环图，其中 K 个顶点上放置了石子，两名玩家轮流行动，按以下规则操作石子：
>
> 移动图上所有还能够移动的石子；
>
> 无法移动石子的一方出局。双方均采用最优策略，询问谁会获胜。

定义 $step$ 为某一局游戏至多需要经过的回合数。

以下局面先手必胜：$step$ 为奇数 。

威佐夫博弈

> 有两堆石子，给出每一堆的石子数量，两名玩家轮流行动，每人每次任选以下规定的一种操作石子：
>
> - 任选一堆，取走正整数颗石子；
> - 从两队中同时取走正整数颗石子。
>
> 拿到最后一颗石子的一方获胜。双方均采用最优策略，询问谁会获胜。

以下局面先手必败：

$\mathbf{(1,2),(3,5),(4,7),(6,10),\dots}$ 具体而言，每一对的第一个数为此前没出现过的最小整数，第二个数为第一个数加上 $\mathbf{1,2,3,4,\dots}$ 。

更一般地，对于第 $\boldsymbol{k}$ 对数，第一个数为 $\boldsymbol{First_k = \left\lfloor \frac{k*(1+\sqrt{5})}{2} \right\rfloor}$ ，第二个数为 $\boldsymbol{Second_k = First_k + k}$ 。

其中，在两堆石子的数量均大于 10^9 次时，由于需要使用高精度计算，我们需要人为定义 $\frac{1+\sqrt{5}}{2}$ 的取值为 $lorry = 1.618033988749894848204586834$ 。

```cpp
const double lorry = (sqrt(5.0) + 1.0) / 2.0;
//const double lorry = 1.618033988749894848204586834;
void Solve() {
    int n, m; cin >> n >> m;
    if (n < m) swap(n, m);
    double x = n - m;
    if ((int)(lorry * x) == m) cout << "lose\n";
    else cout << "win\n";
}
```

斐波那契博弈

> 有一堆石子，数量为 N ，两名玩家轮流行动，按以下规则取石子：
>
> 先手第1次可以取任意多颗，但不能全部取完，此后每人取的石子数不能超过上个人的两倍，拿到最后一颗石子的一方获胜。
>
> 双方均采用最优策略，询问谁会获胜。

当且仅当 N 为斐波那契数时先手必败。

```
int fib[100] = {1, 2};
map<int, bool> mp;
void Force() {
  for (int i = 2; i <= 86; ++ i) fib[i] = fib[i - 1] + fib[i - 2];
    for (int i = 0; i <= 86; ++ i) mp[fib[i]] = 1;
}
void Solve() {
    int n; cin >> n;
    if (mp[n] == 1) cout << "lose\n";
    else cout << "win\n";
}
```

树上删边游戏

> 给出一棵 N 个节点的有根树，两名玩家轮流行动，按以下规则操作：
>
> 选择任意一棵子树并删除（即删去任意一条边，不与根相连的部分会同步被删去）；
>
> 删掉最后一棵子树的一方获胜（换句话说，删去根节点的一方失败）。双方均采用最优策略，询问谁会获胜。

结论：当根节点SG值非 1 时先手必胜。

相较于传统SG值的定义，本题的SG函数值定义为：

- 叶子节点的SG值为 $\mathbf{0}$ 。
- 非叶子节点的SG值为其所有孩子节点SG值 $+1$ 的异或和。

```
auto dfs = [&](auto self, int x, int fa) -> int {
    int x = 0;
    for (auto y : ver[x]) {
        if (y == fa) continue;
        x ^= self(self, y, x);
    }
    return x + 1;
};
cout << (dfs(dfs, 1, 0) == 1 ? "Bob\n" : "Alice\n");
```

无向图删边游戏（Fusion Principle 定理）

> 给出一张 N 个节点的无向联通图，有一个点作为图的根，两名玩家轮流行动，按以下规则操作：
>
> 选择任意一条边删除，不与根相连的部分会同步被删去；
>
> 删掉最后一条边的一方获胜。双方均采用最优策略，询问谁会获胜。

- **对于奇环，我们将其缩成一个新点+一条新边；**
- **对于偶环，我们将其缩成一个新点；**
- **所有连接到原来环上的边全部与新点相连。**

此时，本模型转化为“树上删边游戏”。

/END/