异或方程组

解法

发现异或方程组和普通的方程组有几乎相同的形式，只不过把加法改成了异或。解方程组的时候我们应该也只要把加法消元改成异或消元即可。比如有两个式子$x_k\oplus a_{i2}x_{k+1}\oplus\ldots\oplus a_{im}x_m=b_i$和$x_k\oplus a_{j2}x_{k+1}\oplus\ldots\oplus a_{jm}x_m=b_j$ 我们把两式异或就可消去首项。同时我们可以考虑用bitset优化来减小常数

例题

洛谷p2447

代码

```
#include<bits/stdc++.h>
using namespace std;
const int MAXN = 2001;
inline int read() {
    char c = getchar(); int x = 0, f = 1;
    while(c < '0' || c > '9') {if(c == '-') f = -1; c = getchar();}
    while(c >= '0' && c <= '9') x = x * 10 + c - '0', c = getchar();
    return x * f;
}
int N, M;
bitset<MAXN> b[MAXN];
void Gauss() {
    int ans = 0;
    for(int i = 1; i <= N; i++) {
        int j = i;
        while(!b[j][i] && j < M + 1)
            j++;
        if(j == M + 1) {puts("Cannot Determine"); return ;}
        ans = max(ans, j);
        swap(b[i], b[j]);
        for(int j = 1; j <= M; j++) {
            if(i == j || !b[j][i]) continue;
            b[j] ^= b[i];
        }
    }
    printf("%d\n", ans);
    for(int i = 1; i <= N; i++)
        puts(!b[i][N + 1] ? "Earth" : "?y7M#");
}
int main() {
    N = read(); M = read();
    for(int i = 1; i <= M; i++) {
```

```
        string s; cin >> s;
        b[i][N + 1] = read();
        for(int j = 1; j <= N; j++) b[i][j] = (s[j - 1] == '0' ? 0 : 1);
    }
    Gauss();
    return 0;
}
```