

# Problem-J

## Description

设 $R$ 是集合 $X = \{1, 2, \dots, n\}$ 上的关系，满足以下条件：

- 给定 $m$ 个限制 $x_i, y_i, z_i (x_i \neq y_i)$ ， $z_i = 0$ 表示 $\langle x_i, y_i \rangle \notin R$ ， $z_i = 1$ 表示 $\langle x_i, y_i \rangle \in R$
- $R$ 是自反的
- $R$ 是传递的

求满足条件的 $R$ 的个数。

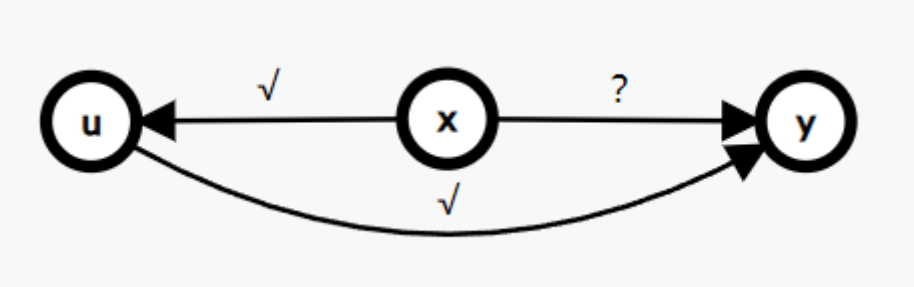
$n \leq 7, m \leq n(n-1)$ ，且答案之和不超过 $3 \times 10^7$ 。

## Solution

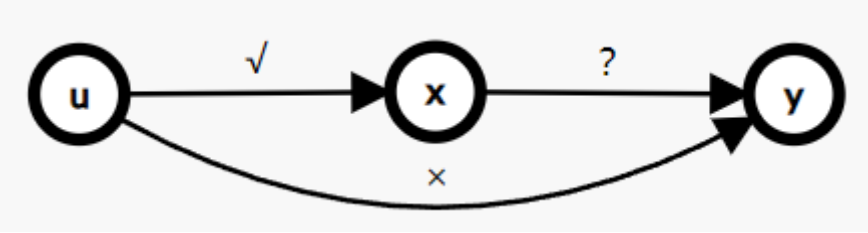
由于 $n$ 很小且题目对答案之和进行了限制，我们考虑搜索。

进行DFS，依次考虑每一对关系 $\langle x, y \rangle$ ，并根据当前情况判断这对关系是应该加入/不能加入/可加可不加（设当前已经确定加入的关系集合为 $S$ ，已经确定不加入的关系集合为 $T$ ）：

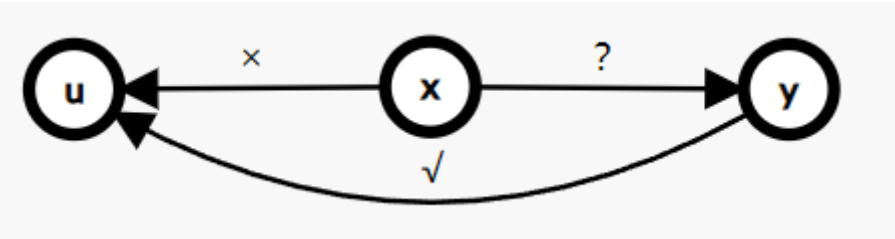
- 若 $\exists u, s.t. \langle x, u \rangle \in S, \langle u, y \rangle \in S$ ，则 $\langle x, y \rangle$ 是必须加入的



- 若 $\exists u, s.t. \langle u, x \rangle \in S, \langle u, y \rangle \in T$ ，则 $\langle x, y \rangle$ 是不能加入的



- 若 $\exists u, s.t. \langle x, u \rangle \in T, \langle y, u \rangle \in S$ ，则 $\langle x, y \rangle$ 是不能加入的



以此为限制进行DFS即可，时间复杂度 $O(\text{可过})$ 。

## Code

```
int n, m, sz, in[9], out[9], IN[9], OUT[9], lmt[9][9], ans, U;
vector<PII> ls;

void DFS(int i) {
    if (i == (int)ls.size()) {
        ++ ans;
        return;
    }

    PII t = ls[i];
    if ((out[t.X] & in[t.Y]) == 0 && lmt[t.X][t.Y] != 1) {
        IN[t.Y] ^= (1 << t.X), OUT[t.X] ^= (1 << t.Y);
        DFS(i + 1);
    }
}
```

```

        IN[t.Y] ^= (1 << t.X), OUT[t.X] ^= (1 << t.Y);
    }
    if (lmt[t.X][t.Y] != 0 && !(in[t.X] & IN[t.Y]) && !(out[t.Y] & OUT[t.X])) {
        in[t.Y] ^= (1 << t.X), out[t.X] ^= (1 << t.Y);
        DFS(i + 1);
        in[t.Y] ^= (1 << t.X), out[t.X] ^= (1 << t.Y);
    }
}

int main() {
    for (int Case = read<int>(); Case --; ) {
        n = read<int>(), m = read<int>();
        for (int i = 0; i < n; ++ i)
            for (int j = 0; j < n; ++ j)
                lmt[i][j] = -1;
        while (m --) lmt[read<int>() - 1][read<int>() - 1] = read<int>();

        ls.clear();
        for (int i = 0; i < n; ++ i)
            for (int j = i + 1; j < n; ++ j)
                ls.PB(MP(i, j)), ls.PB(MP(j, i));
        sz = SZ(ls), ans = 0, U = (1 << n) - 1;
        DFS(0);
        printf("%d\n", ans);
    }
    return 0;
}

```