2-sat

2020-11-23 | 2021-01-05 | OI & ACM | 3
2.4k | 2 分钟

简介

问题大概就是多个 01 变量之间存在 k 元限制，要求在满足限制的情况下构造合法方案

首先我们将所有点拆成两个，i 和 i' 分别表示取 0 和取 1 的情况，边 $u \to v$ 表示如果选 u 就必须选 v

一共有三类限制，其中 a 和 b 的取值为 $0/1$

1. 若 x_i 是 a，则 x_j 一定是 b，反之也成立，那么需要建两条边直接 $x_{i,a} \to x_{j,b}$ 和 $x_{j,b\ xor\ 1} \to x_{i,a\ xor\ 1}$
2. x_i 是 a 和 x_j 是 b 至少满足一个，那么需要连两条边 $x_{i,a\ xor\ 1} \to x_{j,b}$ 和 $x_{i,a} \to x_{j,b\ xor\ 1}$
3. x_i 一定是 a，连边 $x_{i,a\ xor\ 1} \to x_{i,a}$ 这样可以控制在有解的情况下一定选 $x_{i,a}$

注意到这三种限制可以表示出两个变量三种位运算六种结果的所有情况

```
if (s == "AND") {
    if (w) addedge(x, x + n), addedge(y, y + n); //两个变量都一定是 1，限制 3
    else addedge(x + n, y), addedge(y + n, x); // x 是 0 和 y 是 0 至少满足一个，限制
}
if (s == "OR") {
     if (w) addedge(x, y + n), addedge(y, x + n); // x 是 1 和 y 是 1 至少满足一个，
     else addedge(x + n, x), addedge(y + n, y); // 两个变量都一定是 0，限制 3
}
if (s == "XOR") { // 若 x 是 1，则 y 一定是 0；若 x 是 0，则 y 一定是 1
     if (w) addedge(x, y + n), addedge(y, x + n), addedge(x + n, y), addedge(y + n
     else addedge(x, y), addedge(x + n, y + n), addedge(y, x), addedge(y + n, x +
     // 若 x 是 0，则 y 一定是 0；若 x 是 1，则 y 一定是 1
}
```

一个简单的构造解的方法就是 i 和 i' 选择拓扑序较大的

贴个板子

```cpp
#include <iostream>
#include <cstdio>
#include <stack>
#include <queue>
#include <cstring>
#define maxn 1000010
#define maxm 1000010
#define Maxn 2000010
using namespace std;

int n, m;

struct Edge {
    int to, next;
} e[maxm * 2]; int c1, head[Maxn];
inline void add_edge(int u, int v) {
    e[c1].to = v; e[c1].next = head[u]; head[u] = c1++;
}

int dfn[Maxn], low[Maxn], c2, bl[Maxn], scc;
bool ins[Maxn]; stack<int> S;
void tarjan(int u) {
    dfn[u] = low[u] = ++c2; ins[u] = 1; S.push(u);
    for (int i = head[u]; ~i; i = e[i].next) {
        int v = e[i].to;
        if (!dfn[v]) tarjan(v), low[u] = min(low[u], low[v]);
        else if (ins[v]) low[u] = min(low[u], dfn[v]);
    }
    if (low[u] == dfn[u]) {
        int t; ++scc;
        do {
            t = S.top(); S.pop();
            bl[t] = scc; ins[t] = 0;
        } while (t != u);
    }
}

int main() { memset(head, -1, sizeof head);
    ios::sync_with_stdio(false);
    cin.tie(nullptr); cout.tie(nullptr);
```

```
41
42      cin >> n >> m;
43      for (int i = 1; i <= m; ++i) {
44          int x, a, y, b; cin >> x >> a >> y >> b;
45          add_edge(x + (a ^ 1) * n, y + b * n);
46          add_edge(y + (b ^ 1) * n, x + a * n);
47      }
48      for (int i = 1; i <= 2 * n; ++i) if (!dfn[i]) tarjan(i);
49      for (int i = 1; i <= n; ++i)
50          if (bl[i] == bl[i + n]) return cout << "IMPOSSIBLE" << "\n", 0;
51      cout << "POSSIBLE" << "\n";
52      for (int i = 1; i <= n; ++i)
53          cout << (bl[i] > bl[i + n]) << " "; cout << "\n";
54      return 0;
55  }
56
```

---------- 本文结束 感谢阅读 ----------

Tech # 2-sat

‹ Luogu P3834 【模板】可持久化线段树 2（主席树）(整体二分)

manacher ›