Bài toán TWO-SAT

Phạm Lê Quang – FPT University

A. Ví dụ: Excursion

Một nhóm khách du lịch có cơ hội được đi thăm một số thành phố. Mỗi khách du lịch được nêu 2 mong muốn, mỗi mong muốn chỉ thể hiện việc đi thăm hoặc không đi thăm đúng 1 thành phố. Cho phép 2 mong muốn của cùng một người khách là giống nhau hoặc trái ngược nhau. (Ví dụ: tôi muốn đi thăm thành phố A, và tôi không muốn đi thăm thành phố A).

Hãy tính toán xem có thể tạo được một danh sách các thành phố sẽ đi thăm cho nhóm khách này sao cho người khách nào cũng được thỏa mãn ít nhất 1 mong muốn.

Dữ liệu

- Dòng đầu chứa 2 số nguyên n và m thể hiện số khách du lịch và số thành phố. Đánh số các khách du lịch từ 1 đến n và các thành phố từ 1 đến m.
- Dòng thứ i trong số n dòng tiếp theo chứa 2 số nguyên khác 0: w_i và v_i thể hiện mong muốn của khách du lịch thứ i. Với $-m \leq w_i,\ v_i \leq m$. Nếu $w_i > 0$ thể hiện khách du lịch i muốn thăm thành phố w_i, nếu $w_i < 0$ thể hiện khách du lịch i không muốn thăm thành phố $-w_i$. Tương tự cho v_i.

Kết quả

- Nếu không thể tạo được danh sách thành phố sẽ đi thăm thì in ra "NO" (không có dấu ngoặc kép).
- Nếu có thể tạo được danh sách, dòng đầu ghi số nguyên s là số lượng thành phố sẽ đi thăm. Dòng thứ hai ghi đúng s số nguyên theo thứ tự tăng dần, thể hiện các thành phố sẽ đi thăm. Nếu có nhiều đáp án, chỉ cần in ra một trong số đó.

Giới hạn

- $1 \leq n \leq 20000.$
- $1 \leq m \leq 8000.$

Ví dụ

Dữ liệu	Kết quả
3 4 1 -2 2 4 3 1	4 1 2 3 4

B. Phân tích:

Bài toán yêu cầu chọn ra một tập hợp con của tập gồm m thành phố sao cho tất cả n khách du lịch đều được thoả mãn ít nhất 1 trong 2 mong muốn họ đưa ra. Các mong muốn này đều ở dạng "đi thăm thành phố A" hoặc "không đi thăm thành phố A". Từ đây ta nghĩ đến ý tưởng chuyển bài toán về dạng một biểu thức logic.

Trong ví dụ, với dữ liệu:

```
3 4
1 -2
2 4
3 1
```

Có nghĩa là:

- Có 3 khách du lịch, và 4 thành phố.
- Khách du lịch thứ 1 muốn: "Đi thăm thành phố 1" hoặc "Không đi thăm thành phố 2".
- Khách du lịch thứ 2 muốn: "Đi thăm thành phố 2" hoặc "Đi thăm thành phố 4".
- Khách du lịch thứ 3 muốn: "Đi thăm thành phố 3" hoặc "Đi thăm thành phố 1".

Ta gọi: a_i là một biến logic biểu thị cho câu "Đi thăm thành phố i" $(1 \leq i \leq m)$. Kí hiệu $\neg a_i$ là biến logic phủ định của a_i, biểu thị cho câu "Không đi thăm thành phố i".

Do a_i là biến logic nên nó có thể nhận 1 trong 2 giá trị $TRUE/FALSE$.

- $a_i = TRUE$ khi và chỉ khi mong muốn "Đi thăm thành phố i" được thoả mãn (tức là chuyến du lịch sẽ đi qua thành phố i).
- $a_i = FALSE$ khi và chỉ khi mong muốn "Đi thăm thành phố i" không được thoả mãn (tức là chuyến du lịch không đi qua thành phố i), suy ra mong muốn "Không đi thăm thành phố i" được thoả mãn, điều này cũng tương đương với việc $\neg a_i = TRUE$.

Trong ví dụ trên, ta có 4 biến logic là a_1, a_2, a_3, a_4. Bây giờ, mong muốn của những người khách du lịch có thể được biểu diễn bởi công thức sau:

$$\begin{cases} C_1 = a_1 \vee \neg a_2 \\ C_2 = a_2 \vee a_4 \\ C_3 = a_3 \vee a_1 \end{cases}$$

Ở đây kí hiệu ∨ là phép toán "$Tuyển$", hay phép "OR". Để bài toán có nghiệm thì, với mỗi khách du lịch thì ít nhất 1 trong 2 mong muốn phải được thoả mãn, điều này tương đương với việc 1 trong 2 biến logic ở hai bên phép ∨ phải nhận giá trị $TRUE$. Từ đây suy ra $C_1 = C_2 = C_3 = TRUE$ (Theo tính chất của phép OR).

Do tất cả khách du lịch tham gia cùng 1 chuyến du lịch nên các điều kiện trên phải thoả mãn đồng thời:

$$C = C_1 \wedge C_2 \wedge C_3 = (a_1 \vee \neg a_2) \wedge (a_2 \vee a_4) \wedge (a_3 \vee a_1)$$

Ở đây kí hiệu $\wedge$ là phép toán “$Hội$”, hay phép “AND”. Để bài toán có nghiệm thì, do $C_1 = C_2 = C_3 = TRUE$ nên theo tính chất của phép AND suy ra $C = TRUE$.

Bài toán trở thành: Hãy gán giá trị $TRUE/FALSE$ cho các biến a_i sao cho biểu thức C có giá trị $TRUE$, hoặc thông báo là không thể làm được.

C. Bài toán TWO-SAT

Cho m biến logic: $a_1, a_2, \dots, a_m$ và 1 biểu thức logic C có dạng:

$$C = (u_1 \vee v_1) \wedge (u_2 \vee v_2) \wedge \dots \wedge (u_n \vee v_n)$$

Trong đó u_i và v_i $(1 \le i \le n)$ được thay bằng biến logic a_j hoặc $\neg a_j$ nào đó. $(1 \le j \le m)$

Hãy gán giá trị $TRUE/FALSE$ cho các biến $a_1, a_2, \dots, a_m$ sao cho biểu thức C nhận giá trị $TRUE$, hoặc thông báo không thể làm được.

I. Xây dựng đồ thị

❖ **Xét đồ thị vô hướng $G = (V, E)$, trong đó:**

$$\begin{cases} V = \{ a_1, \neg a_1, a_2, \neg a_2, \dots, a_m, \neg a_m \} \\ E = \{ (u_1, v_1), (u_2, v_2), \dots, (u_n, v_n) \} \end{cases}$$

Dễ thấy G có $2m$ đỉnh và n cạnh vô hướng.

Mục tiêu của ta là gán cho tất cả các đỉnh của G giá trị $TRUE/FALSE$ để $C = TRUE$. Điều này tương đương với việc phải tìm ra một tập đỉnh $W \subset V$ chứa tất cả các đỉnh mang giá trị $TRUE$ của G khi biểu thức $C = TRUE$ được thoả mãn. Tập đỉnh W sẽ phải thoả mãn các điều kiện sau:

- (1) ***Trong 2 đỉnh a_i và $\neg a_i$ thì có 1 và chỉ 1 đỉnh thuộc W.***
- (2) ***Với mọi cạnh $(u, v) \in E$ thì ít nhất một trong 2 đỉnh u, v phải thuộc W.***

Ta nói rằng a_i và $\neg a_i$ là 2 đỉnh đối lập vì chỉ 1 trong 2 có thể thuộc tập W.

Xét 1 cạnh $(u, v) \in E$ bất kỳ, ta có $u \vee v$. Mặt khác, theo luật logic, ta lại có:

$$u \vee v \Leftrightarrow (\neg u \to v) \Leftrightarrow (\neg v \to u)$$

Từ đây ta xây dựng một đồ thị có hướng G' mới bằng cách:

- Giữ nguyên tập đỉnh V.
- Nếu đồ thị G có cạnh $(u, v) \in E$ thì vẽ 2 cung: $(\neg u \to v)$ và $(\neg v \to u)$ vào đồ thị mới.

Như vậy đồ thị mới sẽ có $2m$ đỉnh và $2n$ cung có hướng.

❖ **Xét đồ thị có hướng $G' = (V, E')$, trong đó:**

$$\begin{cases} V = \{ a_1, \neg a_1, a_2, \neg a_2, \dots, a_m, \neg a_m \} \\ E' = \{ (u \to v) \mid (\neg u, v) \in E \} \end{cases}$$

Nhớ lại, mục tiêu của ta là tìm tập đỉnh $W \subset V$ chứa các đỉnh mang giá trị $TRUE$ thoả mãn (1) và (2). Khi chuyển sang đồ thị G' thì điều kiện (1) không bị ảnh hưởng nên ta giữ nguyên. Trong khi đó, điều kiện (2) chỉ liên quan đến đồ thị cũ G, nên ta cần phải thay đổi cho phù hợp với đồ thị mới G'.

- Nhận xét: Với 1 đỉnh $w \in W$ bất kỳ. Nếu tồn tại 1 cung $(w \to v) \in E'$ thì suy ra $(\neg w, v) \in E$. Mặt khác, do $w \in W$ nên $\neg w \notin W$, từ đây dẫn đến $v \in W$.

Nhận xét trên cho phép ta chọn được những đỉnh để đưa thêm vào tập W nếu biết được những đỉnh đã có trong W. Do đó, ta sẽ gọi G' là đồ thị suy diễn.

Điều kiện (2) sẽ được thay thế bằng điều kiện tương đương sau:

➢ (2') Nếu $w \in W$ và $(w, v) \in E'$ thì $v \in W$.

Bây giờ ta sẽ đi giải bài toán 2-SAT trên đồ thị suy diễn với 2 điều kiện (1) và (2').

II. Giải bài toán TWO-SAT trên Đồ thị suy diễn

Cho m biến logic: $a_1, a_2, \dots, a_m$ và 1 biểu thức logic C có dạng:

$$C = (u_1 \vee v_1) \wedge (u_2 \vee v_2) \wedge \dots \wedge (u_n \vee v_n)$$

Trong đó u_i và v_i $(1 \le i \le n)$ được thay bằng các biến logic a_j hoặc $\neg a_j$ nào đó. $(1 \le j \le m)$

Xây dựng đồ thị có hướng $G = (V, E)$ trong đó:

$$\begin{cases} V = \{ a_j, \neg a_j \mid 1 \le j \le m \} \\ E = \{ (\neg u_i \to v_i), (\neg v_i \to u_i) \mid 1 \le i \le n \} \end{cases}$$

Hãy tìm cách gán giá trị TRUE/FALSE cho $2m$ đỉnh của G sao cho:

❖ **Tập $W \subset V$ gồm tất cả các đỉnh được gán giá trị $TRUE$ thoả mãn 2 điều kiện sau:**

➢ **(1) Trong 2 đỉnh a_i và $\neg a_i$ thì có 1 và chỉ 1 đỉnh thuộc W.**

➢ **(2) Nếu $u \in W$ và $(u \to v) \in E$ thì $v \in W$.**

Kí hiệu:

1. $u \twoheadrightarrow v$: có đường đi từ u đến v.
2. $D(u) = \{ v \mid v \neq u \text{ và } u \twoheadrightarrow v \}$.
3. $A(u) = \{ v \mid v \neq u \text{ và } v \twoheadrightarrow u \}$.
4. $TPLTM$: thành phần liên thông mạnh.
5. $C(u)$: tập tất cả các đỉnh thuộc TPLTM chứa u (tính cả u).

➢ Trước tiên ta quan tâm đến điều kiện (2):

✓ **Nhận xét 1: Với 1 đỉnh $u \in V$ bất kỳ, nếu $u = TRUE$ thì $\forall v \in D(u)$ ta có $v = TRUE$.**

- Chứng minh: Do $v \in D(u)$ nên sẽ tồn tại 1 đường đi từ u đến v, giả sử là $u \to x_1 \to x_2 \to \cdots \to x_k \to v$ thì do $u = TRUE$ nên theo điều kiện (2) sẽ kéo theo $x_1 = TRUE \Rightarrow x_2 = TRUE \Rightarrow \cdots \Rightarrow x_k = TRUE \Rightarrow v = TRUE$.

✓ **Nhận xét 2: Với 1 đỉnh $u \in V$ bất kỳ, nếu $u = FALSE$ thì $\forall v \in A(u)$ ta có $v = FALSE$.**

- Chứng minh: Khi $u = FALSE$, xét v là một đỉnh bất kỳ thuộc $A(u)$. Do $v \in A(u)$ nên $u \in D(v)$. Bây giờ giả sử $v = TRUE$ thì theo nhận xét 1 phải dẫn đến $u = TRUE$ (trái giả thiết). Vậy $v = FALSE$.

✓ **Nhận xét 3: Với 1 đỉnh $u \in V$ bất kỳ, nếu $u = TRUE$ thì $\forall v \in C(u) : v = TRUE$, nếu $u = FALSE$ thì $\forall v \in C(u) : v = FALSE$. Hay nói cách khác, các đỉnh trong cùng một thành phần liên thông mạnh sẽ phải được gán cùng giá trị.**

- Chứng minh: Khi $u = TRUE$ thì $\forall v \in C(u)$ ta có $u \twoheadrightarrow v \Rightarrow v \in D(u) \Rightarrow v = TRUE$. Tương tự, khi $u = FALSE$ thì $\forall v \in C(u)$ ta có $v \twoheadrightarrow u \Rightarrow v \in A(u) \Rightarrow v = FALSE$.

- **Kết luận:**

1. **Do các đỉnh trong cùng một TPLTM sẽ phải được gán cùng giá trị, nên ta có thể gộp các đỉnh này lại thành 1 đỉnh, từ đó thu được đồ thị đặc biệt dạng DAG.**
2. **Trong đồ thị DAG: nếu biết giá trị của một đỉnh $u = TRUE$ thì phải gán giá trị $TRUE$ cho tất cả các đỉnh $v \in D(u)$. Nếu biết giá trị của $u = FALSE$ thì phải gán giá trị $FALSE$ cho tất cả các đỉnh $v \in A(u)$.**

➢ Bây giờ ta xét đến điều kiện (1):

✓ **Nhận xét 1: Hai đỉnh a_i và $\neg a_i$ phải được gán giá trị khác nhau. Như đã nói ở trên, ta gọi 2 đỉnh này là 2 đỉnh đối lập.**

- Chứng minh: Nếu $a_i = TRUE$ thì $a_i \in W \Rightarrow \neg a_i \notin W \Rightarrow \neg a_i = FALSE$. Nếu $a_i = FALSE$ thì $a_i \notin W \Rightarrow \neg a_i \in W \Rightarrow \neg a_i = TRUE$.

✓ **Nhận xét 2: Nếu hai đỉnh a_i và $\neg a_i$ thuộc cùng 1 TPLTM thì bài toán vô nghiệm (không có cách gán giá trị nào thoả mãn).**

- Chứng minh: Ta biết rằng nếu a_i và $\neg a_i$ thuộc cùng 1 TPLTM thì chúng phải được gán cùng giá trị, điều này trái với nhận xét 1. Vì thế không có cách gán giá trị nào thoả mãn, bài toán vô nghiệm.

✓ **Nhận xét 3: Nếu không có TPLTM nào chứa 2 đỉnh đối lập thì bài toán luôn có lời giải.**

- Chứng minh: Ta sẽ chỉ ra một cách gán giá trị cho các đỉnh trong thuật toán dưới đây:

❖ Thuật toán:

1. **Trên đồ thị $G = (V, E)$, tìm các thành phần liên thông mạnh $C_1,\ C_2, \dots, C_k$.**
2. **Nếu có hai đỉnh đối lập trong cùng 1 TPLTM thì thông báo bài toán vô nghiệm và kết thúc.**
3. **Gộp các đỉnh thuộc cùng một TPLTM, xây dựng đồ thị DAG $G_c(V_c,\ E_c)$:**

$$\begin{cases} V_c = \{\, C_i \mid 1 \le i \le k \,\} \\ E_c = \{\, (C_i \to C_j) \mid \exists u, v \in V: u \in C_i, v \in C_j \text{ và } (u \to v) \in E \,\} \end{cases}$$

   Ta định nghĩa 2 đỉnh đối lập trên G_c như sau:

$$C_i = \neg \mathrm{C_j} \ \Leftrightarrow \exists \mathrm{u} \in \mathrm{V}: \mathrm{u} \in \mathrm{C_i} \text{ và } \neg \mathrm{u} \in \mathrm{C_j}$$

4. **Trên đồ thị G_c:**
   - **Khởi tạo tất cả các đỉnh đều chưa được gán giá trị.**
   - **Sắp xếp các đỉnh theo thứ tự Topo vào danh sách $L[1\,..\,k]$.**
   - **Duyệt các đỉnh lần lượt theo thứ tự "Topo ngược" $(u = L[k] \longrightarrow L[1])$:**
     - ➢ **Nếu đỉnh $u \in V_c$ đang duyệt chưa được gán giá trị thì gán $u = TRUE$.**
     - ➢ **Gán giá trị $FALSE$ cho $\neg \mathrm{u}$ và $\forall v \in V_c: v \in A(\neg \mathrm{u})$.**
5. **Từ cách gán giá trị cho các đỉnh trong G_c suy ra giá trị của các đỉnh trong G.**

Để chứng minh tính đúng đắn của thuật toán trên, ta cần chứng minh các bổ đề dưới đây:

1. Bổ đề:

Bổ đề 1: Khi 1 đỉnh $u \in V_c$ được gán giá trị $TRUE$ thì $\forall v \in D(u) \subset V_c$, đều đã được gán giá trị $TRUE$ từ trước.

- Chứng minh: Theo thuật toán, ta biết rằng cho đến trước khi u được gán giá trị $TRUE$ thì nó chưa hề được gán giá trị.

  Bây giờ, giả sử $\exists v \in D(u)$ mà chưa được gán giá trị $TRUE$ từ trước thì sẽ xảy ra 2 trường hợp:
  - Trường hợp 1: v chưa được gán giá trị.
    Do $v \in D(u)$ nên $u \twoheadrightarrow v$, nên theo phép sắp xếp Topo thì u phải đứng trước v trong danh sách $L[\,]$. Mặt khác, v chưa được gán giá trị thì theo phép duyệt của thuật toán, v phải đứng trước u trong danh sách $L[\,]$. Hai điều này mâu thuẫn nhau.
  - Trường hợp 2: v đã được gán giá trị $FALSE$.
    Do $v \in D(u)$ nên $u \in A(v)$, nên một khi v đã được gán giá trị $FALSE$ thì theo thuật toán, u cũng đã được gán giá trị $FALSE$. Điều này trái với giả thiết là u chưa được gán giá trị.

  Hai trường hợp đều dẫn đến mâu thuẫn chứng tỏ giả sử không xảy ra. Bổ đề được chứng minh.

Bổ đề 2: Khi 1 đỉnh $u \in V_c$ đã được gán giá trị thì giá trị của nó sẽ không bị thay đổi nữa. (Không bị chuyển từ $FALSE$ thành $TRUE$ hoặc từ $TRUE$ thành $FALSE$).

- Chứng minh: Ta sẽ chứng minh không xảy ra cả 2 trường hợp trên:
  - Trường hợp 1: Không bị chuyển từ $FALSE$ thành $TRUE$.
    Dễ thấy, theo thuật toán, ta chỉ gán giá trị $TRUE$ cho 1 đỉnh khi đỉnh đó chưa được gán giá trị. Cho nên nếu u đã được gán giá trị $FALSE$ thì sẽ không bị gán lại thành $TRUE$.

- Trường hợp 2: Không bị chuyển từ $TRUE$ thành $FALSE$.
  Theo thuật toán, ta chỉ gán giá trị $FALSE$ cho 1 đỉnh u nếu như hoặc $\neg$u vừa được gán giá trị $TRUE$, hoặc $\exists v \in D(u) \subset V_c$ mà $\neg v$ vừa được gán giá trị $TRUE$ dẫn đến $\text{v} = FALSE \Rightarrow u = FALSE$.
  Cả 2 trường hợp này đều không thể xảy ra, vì: thứ nhất, do u đang mang giá trị $TRUE$ (theo giả thiết) tức là $\neg$u đã mang giá trị $FALSE$ rồi (theo thuật toán) nên không thể được gán lại thành $TRUE$ (đã c/m ở trên); thứ hai, cũng do u đang mang giá trị $TRUE$ nên theo bổ đề 1 thì $\forall v \in D(u)$ đều đã được gán giá trị $TRUE$, tức là $\forall v \in D(u)$ thì $\neg v$ đã mang giá trị $FALSE$ rồi nên sẽ không thể được gán lại thành $TRUE$.

Hai trường hợp đều không xảy ra. Bổ đề được chứng minh.

Hơn nữa, ta dễ dàng chứng minh được rằng 2 trường hợp còn lại (chuyển từ TRUE thành TRUE, và từ FALSE thành FALSE) cũng không thể xảy ra. Tức là ta có thể kết luận mỗi đỉnh của đồ thị G_c chỉ được gán giá trị đúng 1 lần.

- ***Bổ đề 3: Trên đồ thị G, nếu $u \twoheadrightarrow v$ thì $\neg v \twoheadrightarrow \neg u$.. Với $u, v \in V$.***
- Chứng minh: Do $u \twoheadrightarrow v$ ta có đường đi $u \to x_1 \to x_2 \to \cdots \to x_k \to v$. Mặt khác, theo cách xây dựng đồ thị G ta biết rằng nếu G có cung $(u \to v)$ thì sẽ có cung $(\neg \text{v} \to \neg \text{u})$. Nên từ đường đi trên ta suy ra G có đường đi $\neg v \to \neg x_k \to \cdots \to \neg x_2 \to \neg \text{x}_1 \to \neg \text{u}$, vậy nên $\neg \text{v} \twoheadrightarrow \neg \text{u}$. Bổ đề được chứng minh.

- ***Bổ đề 4: Trên đồ thị G_c, nếu $C(u) \twoheadrightarrow C(v)$ thì $C(\neg v) \to C(\neg u)$. Với u, v là 2 đỉnh của G và C(u) là đỉnh của G_c đại diện cho TPLTM chứa u.***
- Chứng minh: Dễ thấy nếu $\text{C(u)} \twoheadrightarrow \text{C(v)}$ thì trên đồ thị G ta sẽ có $\text{u} \twoheadrightarrow \text{v}$, theo bổ đề 3 dẫn đến $\neg \text{v} \twoheadrightarrow \neg \text{u}$, từ đây suy ra $\text{C}(\neg \text{v}) \to \text{C}(\neg \text{u})$.

2. Chứng minh:

Giả sử các đỉnh của G được gán giá trị theo thuật toán nêu trên, ta sẽ chứng minh tập W gồm các đỉnh mang giá trị $TRUE$ của G thoả mãn cả 2 điều kiện của bài toán:

- ***Chứng minh W thoả mãn (1): Trong 2 đỉnh a_i và $\neg a_i$ thì có 1 và chỉ 1 đỉnh thuộc W.***
  Theo phép duyệt của thuật toán, ta dễ dàng thấy rằng mọi đỉnh của đồ thị G_c chắc chắn sẽ đều được gán 1 trong 2 giá trị $TRUE/FALSE$ (không có đỉnh nào không được gán). Điều này cũng tương đương với việc mọi đỉnh của đồ thị G cũng đều được gán giá trị. Do đó, ta chỉ cần chứng minh a_i và $\neg \text{a}_\text{i}$ không được gán cùng giá trị. Hay điều này tương đương với việc chứng minh $C(a_i)$ và $C(\neg \text{a}_i)$ của đồ thị G_c không được gán cùng giá trị.

- ✓ **Chứng minh $C(a_i)$ và $C(\neg \text{a}_i)$ không cùng mang giá trị $TRUE$:**
  Dễ thấy theo thuật toán, nếu một trong 2 đỉnh $C(a_i)$ và $C(\neg \text{a}_i)$ được gán giá trị $TRUE$ thì đỉnh còn lại lập tức được gán là $FALSE$. Theo ***bổ đề 2*** ta biết rằng các giá trị đã gán này không bị thay đổi cho đến cuối thuật toán. Do vậy 2 đỉnh $C(a_i)$ và $C(\neg \text{a}_i)$ không thể cùng mang giá trị $TRUE$.

- ✓ **Chứng minh $C(a_i)$ và $C(\neg \text{a}_i)$ không cùng mang giá trị $FALSE$:**

Trong 2 đỉnh $C(a_i)$ và $C(\neg a_i)$ sẽ có 1 đỉnh được gán giá trị $FALSE$ trước, giả sử đó là $C(a_i)$. Theo thuật toán, có 2 trường hợp dẫn đến việc $C(a_i)$ được gán giá trị $FALSE$: một là khi $C(\neg \mathrm{a_i})$ được gán là $TRUE$, hai là khi tồn tại một đỉnh $C(v_1) \in D(\mathrm{C(a_i)})$ mà $C(\neg v_1)$ bị gán là $TRUE$ dẫn đến $C(v_1) = FALSE \Rightarrow \mathrm{C(a_i) = FALSE}$

Nếu trường hợp thứ nhất xảy ra thì ta thấy hiển nhiên 2 đỉnh $C(a_i)$ và $C(\neg a_i)$ không cùng mang giá trị $FALSE$. Nếu trường hợp thứ hai xảy ra thì do $C(v_1) \in D(C(\mathrm{a_i}))$ nên $C(a_i) \twoheadrightarrow C(v_1)$, theo bổ đề 4 suy ra $C(\neg \mathrm{v_1}) \twoheadrightarrow \mathrm{C(\neg a_i)}$. Mà $C(\neg v_1) = TRUE$ nên $\mathrm{C(\neg a_i) = TRUE}$. Vậy $C(a_i)$ và $C(\neg a_i)$ không cùng mang giá trị $FALSE$.

- ***Chứng minh W thoả mãn (2): Nếu u ∈ W và $(u \to v) \in E$ thì v ∈ W. Với u, v là 2 đỉnh của đồ thị G.***
- ✓ Nếu $u \in W$ thì $u = TRUE \Rightarrow C(u) = TRUE$. Lại có $(u \to v) \in E$ nên $C(u) \twoheadrightarrow C(v) \Rightarrow C(v) \in D(C(u))$, theo bổ đề 1 thì do $C(u) = TRUE$ dẫn đến $C(v) = TRUE$ suy ra $v = TRUE$, tức là $v \in W$.

Như vậy bài toán đã được chứng minh. Tiếp theo ta sẽ phân tích độ phức tạp của thuật toán trên.

3. Độ phức tạp thuật toán

Ta hãy xem xét lại 5 bước của thuật toán:

Bước	Phân tích	Độ phức tạp
1	Đồ thị $G = (V, E)$ có $2m$ đỉnh và $2n$ cạnh. Thuật toán tìm các TPLTM trên đồ thị này có độ phức tạp $O(2m + 2n)$.	$O(2m + 2n)$
2	Xét lần lượt m cặp đỉnh đối lập để kiểm tra xem chúng có nằm trong cùng 1 TPLTM hay không. Việc xác định số hiệu TPLTM của 1 đỉnh chỉ mất $O(1)$ nhờ việc tìm kiếm sẵn từ bước 1. Do vậy độ phức tạp của bước này là $O(m)$.	$O(m)$
3	Việc gộp các đỉnh của đồ thị G để tạo đồ thị G_c được thực hiện bằng cách duyệt qua tất cả các cạnh của đồ thị G và từ đây xây dựng cung cho đồ thị mới. Độ phức tạp cho các thao tác này là $O(2m + 2n)$.	$O(2m + 2n)$
4	Đồ thị G có k TPLTM tương ứng với k đỉnh của đồ thị G_c. Dễ thấy $k \le 2m$. Thao tác khởi tạo và sắp xếp Topo có thể thực hiện trong $O(k) = O(2m)$. Thao tác duyệt và gán giá trị cho k đỉnh cũng thực hiện trong $O(k) = O(2m)$ vì mỗi đỉnh chỉ được gán giá trị đúng 1 lần (bổ đề 2). Độ phức tạp chung cho bước này là $O(2m + 2m)$.	$O(2m + 2m)$
5	Từ giá trị của các đỉnh của G_c, dễ dàng bằng 1 phép duyệt qua $2m$ đỉnh để gán giá trị cho các đỉnh của G. Độ phức tạp $O(2m)$.	$O(2m)$

Nhìn bảng trên dễ thấy độ phức tạp của cả thuật toán là O(11m + 4n) hay O(m + n).

D. Một số bài luyện tập

1. Colorful Decoration

Bạn đang chuẩn bị trang trí một chiếc bảng trắng bằng các trang giấy màu hình vuông. Mỗi trang giấy có thể có kích thước khác 0 tuỳ ý. Bạn đã chọn được N màu, đánh số từ 0 đến $N-1$, và bạn sẽ dùng đúng 1 trang giấy mỗi màu.

Cho 4 dãy số nguyên X_a, Y_a, X_b, Y_b, mỗi dãy có đúng N phần tử. Trang giấy với màu i phải được đặt sao cho tâm của nó có toạ độ $(X_a[i], Y_a[i])$ hoặc $(X_b[i], Y_b[i])$. Các cạnh của trang giấy phải song song với các trục toạ độ và không có 2 trang giấy đặt đè lên nhau. (Hai hình vuông được coi là đè lên nhau nếu giao điểm của chúng có diện tích khác 0).

Hãy tính kích thước lớn nhất có thể của trang giấy nhỏ nhất trong số N trang. Kích thước của một trang giấy hình vuông là độ dài cạnh của nó. Nếu không thể đặt được tất cả N trang giấy thì in ra 0.

Dữ liệu

- Dòng đầu ghi số nguyên N.
- Dòng 2 ghi N số nguyên thể hiện dãy số X_a.
- Dòng 3 ghi N số nguyên thể hiện dãy số Y_a.
- Dòng 4 ghi N số nguyên thể hiện dãy số X_b.
- Dòng 5 ghi N số nguyên thể hiện dãy số Y_b.

Kết quả

- Gồm 1 dòng duy nhất chứa một số nguyên là kết quả của bài toán, hoặc số 0 nếu vô nghiệm.

Giới hạn

- $2 \leq N \leq 50$.
- $0 \leq X_a[i], Y_a[i], X_b[i], Y_b[i] \leq 1{,}000{,}000{,}000$.

Ví dụ

Dữ liệu	Kết quả
3 10 0 7 0 19 6 20 10 25 20 35 25	19
2 464 20 464 10 464 3 464 16	461
4 0 0 1 1 0 0 1 1 1 1 0 0 1 1 0 0	0

2. Election

Sau nhiều năm chiến tranh liên miên giữa các Đảng phái, nước X rơi vào tình trạng đói nghèo, người dân khổ cực trăm bề. Nhận thức được việc nếu tiếp tục kéo dài chiến tranh sẽ càng bất lợi cho đất nước, các Đảng trong nước X đã quyết định họp bàn nhau lại, bỏ qua hiềm khích chung để xây dựng lại đất nước.

Việc làm đầu tiên sẽ là họp để chọn ra các vị đại biểu để lập nên Quốc Hội. Mỗi Đảng đã chọn ra 2 gương mặt tiêu biểu nhất cho Đảng của mình để ứng cử vào Quốc Hội. Tuy nhiên trong số các vị đại biểu của các Đảng này thì có một số vị vì lý do cá nhân trong chiến tranh nên rất căm thù nhau (ví dụ như ông A của Đảng P ghét ông B của Đảng Q, …).

Vì lý do chính trị mà trong Quốc Hội mỗi Đảng chỉ được phép có một người mà thôi. Ngoài ra để đảm bảo Quốc Hội làm việc 1 cách công minh thì các vị đại biểu Quốc Hội phải được chọn ra sao cho đảm bảo không có ai thù ghét ai cả, nếu không rất có thể chiến tranh sẽ lại nổ ra.

Bạn là một người yêu chuộng hoà bình đồng thời là 1 lập trình viên siêu hạng. Hãy xem xét xem liệu có 1 cách tổ chức Quốc Hội sao cho thoả mãn được các yêu cầu đề ra hay không?

Dữ liệu

- Dòng 1: 2 số nguyên N và M tương ứng là số Đảng và số mối quan hệ thù ghét nhau giữa các ứng viên của các Đảng. Các Đảng được đánh số từ 1 đến N. Hai ứng viên của Đảng i sẽ có số hiệu là $i * 2 - 1$ và $i * 2$.
- M dòng tiếp theo mỗi dòng gồm 2 số nguyên u, v cho biết người u và người v ghét nhau $(1 \leq u < v \leq 2N)$.

Kết quả

- Dòng 1: Ghi 0 nếu không có phương án thoả mãn và 1 nếu có phương án thoả mãn.
- Nếu dòng 1 là 1 thì dòng thứ 2 ghi ra N số nguyên là số hiệu của các thành viên được chọn vào Quốc Hội.

Giới hạn

- $1 \leq N \leq 8000$.
- $1 \leq M \leq 20000$.

Ví dụ

Dữ liệu	Kết quả
3 2 1 3 2 4	1 1 4 5

E. Tham khảo:

1. BOI 2001 Task and Solution: Polish Olympiad in Informatics – 7th Baltic Olympiad in Informatics.
2. Chuyên tin tự giới thiệu: ĐHKHTN – ĐHQGHN.
3. Topcoder Problems and Editorials: www.topcoder.com/tc.