LÍ THUYẾT TRÒ CHƠI

Chuyên đề DHBB 2017 – Tin học

Impartial Combinatorial Games

Các trò chơi đối kháng giữa hai người đã được hình thành từ lâu trong đó những người chơi luôn cố gắng tìm mọi cách để mình giành được phần thắng. Về lí thuyết khả năng thắng thua của các trò chơi là hoàn toàn tiền định. Chính xác hơn, nếu một trò chơi cho trước vị trí ban đầu thì kết quả tốt nhất mà người chơi đầu tiên đạt được là có thể tính toán được (giả thiết cả hai người chơi đều chơi tối ưu). Tuy nhiên do khối lượng tính toán quá lớn nên khả năng thực hiện nước đi tối ưu đối với một số trò chơi là không tưởng. Do vậy cho đến nay, chỉ một số lượng nhỏ trò chơi đối kháng được giải quyết trọn vẹn. Bài viết này giới thiệu hai cách tiếp cận trò chơi đối kháng hai người: tìm kiếm mini-max và hàm grundy

1. Mô hình đồ thị của trò chơi đối kháng

Cho đồ thị định hướng $G = (V, E)$ (đồ thị G có tập đỉnh V, tập cạnh E), ta định nghĩa $E(v) = \{ u | (v, u) \in E \}$ tức là tập đỉnh đến được từ u

Một trò chơi hai người tương ứng với một đồ thị định hướng $G = (V, E)$ trong đó mỗi trạng thái chơi tương ứng với một đỉnh của đồ thị, hàm $E(u)$ là qui tắc chơi theo nghĩa $E(u)$ chứa các đỉnh hay trạng thái chơi mà từ u có thể đi đến. Hai người luân phiên nhau thực hiện lượt đi, ở thế chơi u người chơi chỉ có thể đi sao cho nước v nhận được thoả mãn $v \in E(u)$. Trò chơi kết thúc khi không thể đi tiếp được nữa. (Thông thường thì người không thể đi tiếp là người thua cuộc).

Ta phân chia các trò chơi thành hai loại: trò chơi lựa chọn một đối tượng và trò chơi lựa chọn nhiều đối tượng tùy theo khả năng phân rã đồ thị tương ứng của trò chơi thành các đồ thị con độc lập.

2. Trò chơi lựa chọn một đối tượng

Loại trò chơi lựa chọn một đối tượng có thể hiểu trực quan là các nước đi thực hiện trên cùng một đối tượng (nên G không phân rã thành các đồ thị con độc lập được).

Ví dụ 1. Trò chơi tổng chính phương

Bắt đầu bằng một số nguyên dương N, hai đấu thủ A, B lần lượt trừ bớt số đó đi một số chính phương dương sao cho phần còn lại không âm. Đấu thủ thua nếu không thực hiện được nước đi trong lượt của mình. Nếu mỗi người đều chơi tối ưu thì liệu người thứ nhất có khả năng thắng?

Phân tích

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
☠	☺	☠	☺	☺	☠	☺	☠	☺	☺	☠	☺	☠	☺	☺

Lượt chơi có $N = 0$ hiển nhiên là thua.

Lượt chơi có $N = 1,4,9,16, \dots$ sẽ chiến thắng ngay sau lượt đi đầu tiên.

Lượt chơi $N = 2$ chỉ có thể trao cho đối thủ $N = 1$ nên chắc chắn thua.

Lượt chơi $N = 3$ có thể trao cho đối thủ $N = 2$ nên chắc chắn thắng.

Lượt chơi $N = 5$ chỉ có thể trao cho đối thủ $N = 1,4$ (chắc chắn thắng) nên chắc chắn thua

…

Bằng cách phân tích tương tự ta tìm được khẳng định chắc chắn cho N bất kì (miễn là có đủ thời gian tính toán).

Để tổng quát hóa cách làm trên, ta đưa ra các khái niệm thế thắng, thế thua

- Khi đến lượt chơi mà không còn nước đi nào nữa là ta đang ở thế thua (gọi là thế thua cơ sở)
- Nếu ở lượt chơi có thể thực hiện một nước đi đưa đối phương vào thế thua thì ta đang ở thế thắng
- Nếu mọi nước đi trong lượt chơi đều đưa đối phương vào thế thắng thì ta đang ở thế thua

Định nghĩa truy hồi trên cho phép ta áp dụng phép tìm nước đi theo chiến lược vét cạn với các trò chơi có số trạng thái đủ nhỏ (dĩ nhiên)

Ngoài ra, do giả thiết hai đấu thủ đều chơi tối ưu nên tính thắng/thua của mỗi trạng thái là tiền định, vậy có thể thực hiện phương pháp loang thông thường để tô màu các trạng thái (quyết định thế thắng/thua)

Nếu trò chơi không chỉ là quyết định thắng/thua mà còn tính điểm các nước đi thì ở mỗi nước đi, đấu thủ luôn mong muốn mình giành được điểm số cao nhất khi kết thúc trò chơi. Tương tự như trên có thể thấy chiến lược thực hiện nước đi sẽ là: chọn nước đi sao cho sau đó với cách đi tốt nhất của đối phương, điểm giành được là lớn nhất có thể. Chính xác hơn, đối phương sẽ cố gắng cực tiểu hóa điểm số của ta, do đó ta cần thực hiện nước đi sao cho điểm số cực tiểu đó lớn nhất có thể. Đây là lí do vì sao cách tiếp cận này có tên là tìm kiếm mini-max.

Ta xét một số bài toán ví dụ

Ví dụ 2. GAME

Một trò chơi đối kháng giữa hai người A và B diễn ra như sau: Hai người luân phiên nhau điều khiển một con tốt theo một số con đường cho trước. Một người có thể di chuyển con tốt từ vị trí u đến v nếu có một đường nối trực tiếp định hướng từ u đến v. Trò chơi kết thúc khi không thể tiếp tục di chuyển. Đấu thủ không thể thực hiện nước đi là người thua cuộc. Hỏi nếu cho trước vị trí ban đầu và danh sách các đường nối thì người đi trước thắng, thua hay hoà? Giả hai người này luôn chơi tối ưu.

Input: GAME.IN

- Dòng đầu ghi số N là số vị trí con tốt có thể đứng, và số M là số đường đi (định hướng) mà con tốt có thể đi (1≤ N ≤ 200, 1 ≤ M ≤ N*(N-1)).
- Dòng thứ hai ghi u là trạng thái bắt đầu.
- M dòng tiếp theo mỗi dòng ghi hai số u, v mô tả một đường đi từ u đến v.

Output: GAME.OUT

- Ghi một số duy nhất 1, 2, hoặc 0 tương ứng với người thắng, thua hay hoà.

Phân tích

- Những vị trí không có đường ra thì chắc chắn sẽ thua.
- Những vị trí nào có một đường ra nối với vị trí chắc chắn thua thì chắc chắn thắng.
- Những vị trí nào tất cả các đường ra đều nối với các vị trí chắc chắn thắng thì chắc chắn thua.
- Những vị trí nào mà trạng thái thắng thua không thể xác định thì là vị trí hoà

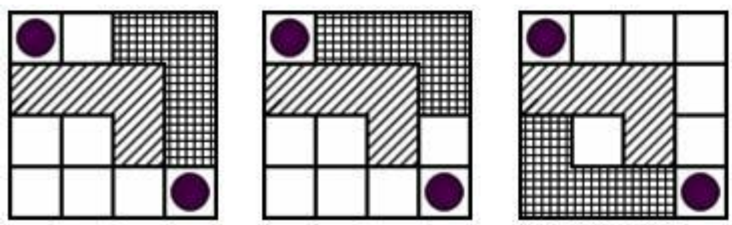
Thuật toán

- Ban đầu coi tất cả các vị trí v đều hoà gán giá trị đỉnh $F[v] = 0$.
- Tìm các vị trí không có đường ra thì gán lại $F[v] = 2$ (vị trí thua).
- Khi thay trạng thái một vị trí từ hoà sang thắng hoặc thua thì kiểm tra các vị trí có đường đi đến nó: Những vị trí u nào có một đường ra nối với vị trí v chắc chắn thua ($F[v] = 2$) sẽ là chắc chắn thắng (thay $F[u] = 1$); Những vị trí u nào tất cả đường ra nối với đều nối với các vị trí v có $F[v] = 1$ (chắc chắn thắng) thì chắc chắn thua (thay $F[u] = 2$).
- Quá trình này ngừng khi không có sự chuyển trạng nào nữa.
- Trò chơi có thể hòa vì sự xuất hiện của chu trình.

Ví dụ 3. LGAME

Cho một bảng kích thước 4*4 ô vuông, trên đó đặt hai thanh thước thợ hình L kích thước 4 ô vuông và hai hình tròn như hình vẽ, các hình này nằm trên bảng và không được đè lên nhau. Hình kẻ ca rô là của người chơi A, hình kẻ sọc của người chơi B. Hai người sẽ chơi luân phiên, tại mỗi nước đi, một người sẽ phải nhấc thanh hình L của mình lên, xoay, lật tuỳ ít và di chuyển đến vị trí mới (khác ít nhất một ô so với vị trí ban đầu), như vậy hình đầu tiên có hai cách di chuyển. Và người chơi có thể thực hiện thêm một bước đi không bắt buộc là di chuyển một ô tròn đến một ô mới.

Trò chơi kết thúc khi không thể di chuyển được nữa, người không thể di chuyển được sẽ thua cuộc. Tuy nhiên, trò chơi vẫn có thể hoà vì trong trạng thái đó cả hai người đều không muốn thua (chính xác hơn là không dồn được đối thủ vào thế thua).



Yêu cầu: Cho một trạng thái trò chơi, hỏi trò chơi đó sẽ kết thúc như thế, (hoà, A thắng hay B thắng, ở đây A là người đi trước)

Input

- Gồm 4 dòng mỗi dòng ghi 4 kí tự, '.' thể hiện ô trống(có 6 ô), 'x' là ô chứa miếng hình tròn (2 ô), '#' biểu thị ô bị miếng hình L của người chơi A đặt lên (có 4 ô), còn lại bốn ô biểu thị ô bị miếng hình L của người chơi B đặt lên.

Output

Có ba trường hợp:

- A thắng: ghi trạng thái sau khi A đi nước đi đầu tiên dẫn đến trạng thái thắng đó.
- A thua: ghi ra xâu "No winning move Losing".
- Hoà: ghi ra xâu "No winning move Draw".

Ví dụ

LGAME.IN	LGAME.OUT
.*** #*.x ###. x...	.*** x*#x ###. ....
...x ###. #*** x..*	No winning move Draw
.### x#*x ***.	No winning move Losing

Phân tích

- Do kích thước bảng là nhỏ nên có thể tính toán được số trạng thái là vào khoảng 18,000. Từ đó có thể xây dựng một hàm băm hợp lí và tiến hành loang để tô màu các trạng thái.

Đôi khi không phải lúc nào cũng có thể lưu được tất cả các trạng thái vì trong một số bài toán số trạng thái rất lớn. Khi đó, thay vì xét thế thắng/thua cho trạng thái hiện thời ta tính thế thắng/thua cho trạng thái tương đương.

Ví dụ 4. Stones

Một trò chơi bốc sỏi diễn ra trên một bảng ngang kích thước 1*N ô vuông. Trên một số ô có đặt một số viên sỏi. Tại một bước đi người chơi cầm một viên sỏi ở một ô và di chuyển viên sỏi sang bên trái một hoặc hai ô với điều kiện là ô di chuyển tới phải không có sỏi và đường di chuyển không được qua ô có sỏi. Người nào không di chuyển được sẽ là người thua cuộc. Cho trước trạng thái ban đầu hỏi người đi trước có bao nhiêu nước đi đầu tiên mà người thứ hai luôn thua, giả thiết cả hai người đều chơi tối ưu.

Input

- Dòng đầu ghi số N($1 \le N \le 50$).
- Dòng thứ hai ghi một xâu gồm N kí tự thể hiện trạng thái lúc bắt đầu trò chơi, '.' thể hiện ô trống, 'X' thể hiện có sỏi (số viên sỏi không vượt quá 10).

Output

- Ghi một số là số nước đi mà có thể thắng.

Ví dụ

STONES.IN	STONES.OUT
3 X.X	1
4 ..XX	0

Phân tích

Nếu coi mỗi trạng thái là một đỉnh đồ thị thì rõ ràng bài toán theo lý thuyết có thể tính được kết quả. Nhưng trên thực tế số trạng thái rất lớn (có thể lên đến C_{50}^{10}).

Vì vậy trước hết cần làm giảm số lượng trạng thái cần xét. Đầu tiên ta thấy trạng thái của người chơi được đặc trưng bởi dãy các số nguyên không âm là số ô tự do ở phía trước mỗi viên sỏi, ví dụ: xâu "...XX.X" thay bởi dãy {3, 0, 1}. Tuy nhiên đó mới chỉ là cách mã hóa trạng thái, ta sẽ xét sự thắng thua của dãy khi lấy đồng dư modul 3 của tất cả các phần tử trong dãy, chẳng hạn thay vì dãy {3,0,1} ta xét dãy {0,0,1}

Chứng minh được hai dãy này là tương đương thắng/thua, thật vậy:

Gọi dãy ban đầu là A, dãy sau khi ước lược là $B = f(A)$ (f là hàm rút gọn).

Vì B là dãy ước lược của A nên với cách đi một nước bất kì từ B đến B' thì A cũng đi có thể đi một nước đến A' (cùng vị trí và số ô) sao cho $f(A') = f(B')$. (I)

Chẳng hạn với dãy B{0,0,1} ở trên sau một nước đi vị trí 3 với số ô đi bằng 1 đến B'{0,0,0} thì với A{3,0,1} cũng đi tại 3 với số ô bằng 1 đến A'{3,0,0}. Lúc đó ta vẫn có: $f(A') = f(B') =$ {0,0,0}.

Vì mọi bước chơi của đối thủ đều nhằm có lợi cho mình nên nếu người chơi thứ nhất thực hiện một nước đi từ A đến A' hòng thay đổi sự thua thành thắng (vốn theo lý thuyết là xác định), tức $f(A)$ thua, $f(A')$ thua mà $B = f(A)$, suy ra B không đi được đến B' (vì $B = f(A)$ suy ra B thua, B' cũng thua) suy ra người chơi đã thực hiện trên một ô có số ô tự do ở đằng trước lớn hơn bằng 3, suy tiếp ra người thứ hai có thể đi tiếp một nước trên cùng ô đó với số ô bằng (3 - số ô người thứ nhất đã đi). Suy ra người thứ nhất vẫn ở vị trí $f(A'')$ thua. (II)

(I)(II) $\Longrightarrow$ hai dãy là tương đương theo kết quả thắng thua.

Ví dụ 5. Trò chơi chuyển đá

Vào một ngày đẹp trời, A nghĩ ra một trò chơi và rủ B cùng tham gia. Có n ô, mỗi ô chứa một số viên đá. Các ô được đánh số từ 0 đến $n - 1$. Để thực hiện một nước đi, A/B chọn 3 ô với chỉ số i, j, k thoả mãn $i < j \leq k$ và ô i chứa ít nhất 1 viên đá, sau đó bỏ đi 1 viên đá ở ô i đồng thời thêm hai viên đá vào ô j và ô k (mỗi ô một viên). Chú ý là j có thể bằng k, và sau mỗi bước tổng số viên đá luôn tăng lên 1. Ai không thể thực hiện nước đi coi như bị thua. A đi trước.

Nhiệm vụ của bạn là xác định xem A có thể chiến thắng hay không? (giả sử B chơi tối ưu). Nếu có thể hãy in ra 3 số i, j, k mô tả nước đi đầu tiên của A. Nếu có nhiều kết quả hãy in ra kết quả có i nhỏ nhất, nếu vẫn có hơn một kết quả chọn kết quả có j nhỏ nhất, nếu vẫn có hơn một kết quả chọn kết quả có k nhỏ nhất.

Input

- Dòng đầu gồm số nguyên n là số ô.
- Dòng thứ hai gồm n số, số thứ i thể hiện số viên đá ở ô i.

Output

- Nếu A thắng thì in ra 3 số i, j, k trên một dòng duy nhất.
- Nếu A thua thì in ra một số –1 duy nhất.

Giới hạn

- $1 \le n \le 15$
- Số viên đá ở một ô không vượt quá 1000.

Ví dụ

INCSTONE.IN	INCSTONE.OUT
5 2 1 1 1 5	0 1 1

Phân tích

Trạng thái tương đương là trạng thái lấy số sỏi mỗi ô mod 2, như vậy có nhiều nhất là 2^{15} trạng thái. Vậy có thể áp dụng kĩ thuật tìm mini-max.

3. Trò chơi lựa chọn nhiều đối tượng

Qua cách tiếp cận trên ta nhận thấy rằng nếu mỗi trạng thái chơi gồm nhiều đỉnh ở trên một đồ thị thì chúng ta không thể thực hiện tìm kiếm mini-max đơn thuần được do số trạng thái rất lớn, phương pháp giải quyết vấn đề này là dùng hàm grundy.

Ví dụ 6. Trò chơi bốc sỏi

Hai người chơi bốc sỏi trên một đống sỏi có n viên, mỗi người đến lượt mình chỉ được bốc không quá k viên. Xác định khả năng thắng của người đi trước nếu cả hai đều chơi tối ưu

Phân tích

Ta khảo sát với $k = 3$

Ta đặc trưng cho số sỏi $n = 0$ đại lượng 0 để chỉ thế thua cơ sở.

Với mỗi số $n > 0$ có nhiều nhất 3 nước đi có thể thực hiện đưa đến các giá trị n_1, n_2, n_3, đặc trưng cho n bởi giá trị nguyên không âm nhỏ nhất chưa xuất hiện trong các giá trị đặc trưng của n_1, n_2, n_3. Với cách làm này ta thu được bảng:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
0	1	2	3	0	1	2	3	0	1	2	3	0	1	2

Dễ nhận thấy rằng nếu n có trị đặc trưng khác 0 thì ta luôn có nước đi để trao cho đối thủ số sỏi có đặc trưng bằng 0. Ngược lại, nếu đặc trưng cho n bằng 0 thì hoặc ta không có nước đi ($n = 0$) hoặc mọi nước đi của ta đều dẫn đến số sỏi có đặc trưng không bằng 0.

Từ đó rút ra thế thua là các trạng thái có số đặc trưng bằng 0 là số đặc trưng cho thế thua cơ sở (*).

Với các trò chơi tương tự cũng có thể tìm số đặc trưng cho các trạng thái theo cách trên và cũng thu được kết luận (*). Đây chính là cơ sở về hàm grundy.

3.1. Khái niệm

Cho đồ thị định hướng và không có chu trình $G = (V, E)$ trong đó V là tập đỉnh, E là tập cạnh.

- Với mỗi đỉnh $u \in V$, ta định nghĩa $E(u) = \{v|(u, v) \in E\}$.
- Một trò chơi hai người được định nghĩa là một đồ thị định hướng $G = (V, E)$ trong đó mỗi thế chơi tương ứng với một đỉnh của đồ thị tổng (có thể gồm một hay nhiều đồ thị thành phần), hàm $E(v)$ là qui tắc chơi, mỗi một đỉnh v được gán với một số thực $r(v)$ là giá trị mà mỗi đấu thủ nhận được trên mỗi nước đi.
- Hai người luân phiên nhau đi, ở thế chơi u người chơi chỉ có thể đi sao cho nước v nhận được thoả mãn $v \in E(u)$, và khi đó người đi nhận được giá trị $r(v)$ tương ứng. Trò chơi kết thúc khi đến lượt chơi mà không thể thực hiện nước đi.
- Nếu trò chơi kết thúc, tuỳ theo tổng giá trị mà mỗi đấu thủ nhận được ta có kết quả là có người thắng hay ván đấu hoà.

Trước hết, ta định nghĩa hàm grundy như sau: Cho đồ thị $G = (V, E)$, hàm grundy trên G là cách gán cho mỗi đỉnh u một số tự nhiên $g(u)$, sao cho $\forall u \in V,\ g(u)$ là số tự nhiên nhỏ nhất không có trong tập S các giá trị hàm grundy của các đỉnh kề với u.

3.2. Các định lý

Ta xét một lớp bài toán trò chơi thoả mãn các tính chất: Đồ thị G tương ứng là đồ thị định hướng không có chu trình và ở đây người thắng cuộc là người thực hiện nước đi cuối cùng trước khi kết thúc trò chơi. Từ đây, nói đến đồ thị G, ta chỉ nói đến đồ thị thỏa mãn tính chất trên.

Định lý 1

Đồ thị DAG có và có duy nhất một hàm grundy.

Chứng minh:

Ta xác định hàm grundy như sau:

- Ban đầu tất cả các đỉnh đều chưa tính hàm grundy.
- Lần lượt thực hiện chừng nào vẫn còn đỉnh chưa tính hàm: Xét đồ thị con G' của G sinh ra bởi tập hợp các đỉnh chưa tính hàm, lấy tất cả các đỉnh v trong G' mà $E'(v)$ tương ứng bằng $\emptyset$, ta xác định hàm grundy cho mỗi đỉnh v đó theo đúng định nghĩa hàm grundy: $S(v) = \{g(u)|u \in E(v)\};\ g(u) = min\{p|(p \in N) \cap !(p \notin S(u))\}$

Chú ý rằng vì G định hướng và không có chu trình nên mọi đồ thị con G' của nó cũng thoả mãn không có chu trình. Mặt khác một đồ thị định hướng không có chu trình thì luôn luôn tìm được đỉnh không có cung đi ra tức là $E' = \emptyset$, như vậy tại mỗi bước luôn xác định hàm grundy cho ít nhất một đỉnh. Ta sẽ chỉ ra rằng thuật toán trên xác định hàm grundy duy nhất, bằng phương pháp qui nạp.

- Tại bước 0, hiển nhiên các đỉnh được chọn có giá trị hàm grundy $= 0$, vì các đỉnh v này đều có $E(v) = 0$ nên theo định nghĩa hàm grundy, đây là cách cho duy nhất với các đỉnh này.
- Giả sử tại một bước $k \geq 1$, tất cả các đỉnh đã tính hàm đều thỏa hàm tương ứng là xác định duy nhất.
- Khi đó, các đỉnh được xác định hàm trong bước này không có cung nối đến các đỉnh chưa có hàm, do đó với mỗi đỉnh v được xác định hàm trong bước này, tập $E(v)$ chỉ gồm các đỉnh đã có hàm được xác định duy nhất, do đó theo định nghĩa hàm grundy, thì hàm $g(v)$ sẽ được xác định duy nhất.
- Như vậy, đến bước $k + 1$, tất cả những đỉnh đã được xác định hàm grundy thì đều được xác định một cách duy nhất. Định lý được chứng minh.

Định lý 2

Đồ thị $\boldsymbol{G}$ nhận $\boldsymbol{g}$ là hàm grundy tương ứng, khi đó nếu đấu thủ đến lượt chơi ở trạng thái có hàm grundy khác 0 thì đấu thủ đó luôn luôn có cách chơi để không thua.

Chứng minh

Tại thế chơi khác 0, ta luôn có cách đưa về thế chơi 0, vì nếu không thì thế chơi đó phải là 0 theo đúng định nghĩa hàm grundy. Đối phương khi đã ở thế 0 chỉ có thể không đi tiếp được nữa, hoặc đi đến một thế chơi khác 0 cho ta. Như vậy, đến lượt ta luôn là thế chơi khác 0, tức là ta sẽ không thua, trong trường hợp đồ thị hữu hạn không có chu trình thì ta sẽ chắc thắng.

Hệ quả: Với đồ thị hữu hạn ta luôn có thể xác định được người chiến thắng.

Như vậy vấn đề ở đây là phải xác định các hàm grundy. Cần phải chú ý rằng nếu đồ thị không là DAG thì có thể có nhiều hàm grundy, nên các kết quả trên không đúng.

Xét định nghĩa phép cộng các đồ thị:

Cho hai đồ thị $G_1 = (V_1, E_1)$; $G_2 = (V_2, E_2)$ khi đó, $G = G_1 + G_2$ là một đồ thị $G(V, E)$ có: $V = V_1 \times V_2$ và $\big((x_1, x_2), (y_1, y_2)\big) \in E \ \Leftrightarrow \big((x_1 = y_1) \cap (x_2, y_2) \in E_2\big) \cup \big((x_2 = y_2) \cap (x_1, y_1) \in E_1\big)$.

Tổng quát ta định nghĩa bằng qui nạp $G = G_1 + \cdots + G_n = (G_1 + G_2 + \cdots + G_{n-1}) + G_n$.

Định lý 3

Cho $\boldsymbol{G_1}, \boldsymbol{G_2}, \dots, \boldsymbol{G_n}$ lần lượt nhận các hàm grundy $\boldsymbol{g_1}, \boldsymbol{g_2}, \dots, \boldsymbol{g_n}$ tương ứng. Khi đó: $\boldsymbol{G} = \boldsymbol{G_1} + \cdots + \boldsymbol{G_n}$ sẽ nhận hàm grundy $\boldsymbol{g}$ thoả mãn:

$$\boldsymbol{g(v_1, v_2, \dots, v_n) = g_1(v_1)\ XOR\ g_2(v_2)\ XOR \dots XOR\ g_n(v_n)}$$

Chứng minh:

Vì G cũng thoả mãn các tính chất của $G_1, G_2 \dots$ nên nó có hàm grundy duy nhất. Để chứng minh công thức trên là hàm grundy , ta chứng minh nó đúng với từng đỉnh theo thứ tự như trong định lý 1: Khi xét một đỉnh v thì các đỉnh trong tập $E(v)$ đều đã thoả mãn.

Ta có hai tính chất sau:

1. Cách cho hàm g thoả mãn $g(v)$ khác $g(u)$ với mọi $u \in E(v)$ vì nếu ngược lại $g(u) = g(v)$: u và v chỉ khác nhau 1 thành phần trong biểu diễn $(x_1, x_2, \dots, x_n)$, giả sử tại thành phần thứ $k \Longrightarrow g(u_k) = g(v_k)$, mâu thuẫn vì $u_k \in E_k(v_k)$.
2. Hàm g là nhỏ nhất trong tập giá trị thoả mãn tính chất 1: Giả sử $l < g(v)$ thỏa mãn, khi đó trong biểu diễn nhị phân, gọi là vị trí đầu tiên sai khác giữa $g(v)$ và l dễ thấy của $g(v)$ là 1 còn l là 0. Vì $g(v)$ vị trí $i = 1$ nên tồn tại thành phần, giả sử j có $g_j(v_j)$ trong biểu diễn nhị phân vị trí thứ i bằng 1 nên tồn tại $v_j' \in E_j(v_j)$ sao cho $g_j(v_j') = g_j(v_j)\ XOR\ 2^i$. Khi đó: $v' = (v_1, v_2, \dots, v_j', \dots, v_n)$ có $l = g(v')$ không thoả mãn hàm grundy vì $v' \in E(v)$.

Cách khác: Giả sử $l < g(v)$ thoả mãn suy ra tồn tại một số vị trí xảy ra sự sai khác về cách biểu diễn nhị phân ở vị trí: $i_n < i_{n-1} < \cdots < i_1$. Trong đó i_n của $g(v)$ là 1 còn i_n của là 0. Lấy $g(v_i)$ sao cho bit thứ i_n là 1. Biến đổi $g(v_i)$ thành $g(v_i')$ sao cho các vị trí $i_n, i_{n-1}, \dots, i_1$ đảo ngược. (Luôn biến đổi được vì $g(v_i) < g(v_i')$). Lúc đó $g(v') = l$. Điều vô lý này chứng tỏ không tồn tại $l < g(v)$

Từ hai tính chất trên khẳng định được g trên là hàm grundy của G.

Một ứng dụng rất quan trọng của hàm grundy là giải các đồ thị hợp bằng công thức cộng đồ thị trên. Ta xét một số bài toán cụ thể.

Ví dụ 7. Trò chơi NIM

Hai người chơi trò chơi bốc sỏi với n đống sỏi với số sỏi tương ứng là $m_1, m_2, \dots, m_n$. Mỗi lượt đi người chơi chọn một đống sỏi và bốc ra một số lượng tùy ý sỏi trong đống đó (có thể bốc hết). Trò chơi kết thúc khi không còn sỏi để bốc và người chơi không thực hiện được nước đi thua cuộc. Hãy xác định khả năng thắng của người đi trước và chỉ ra một nước đi đầu tiên của người đó.

Phân tích

Rõ ràng đây là trò chơi lựa chọn nhiều đối tượng. Mỗi đối tượng là một đống sỏi có thể tách thành một đồ thị con.

Kết quả trên một đồ thị con hay áp dụng trò chơi với một đống sỏi là quá hiển nhiên vì người đi trước chỉ việc bốc tất cả số sỏi trong đống.

Để có kết quả của đồ thị hợp ta cần hàm grundy cho mỗi đồ thị con. Dễ thấy chúng đều là $g_i(m_i) = m_i$

Vậy hàm grundy của đồ thị hợp là $t = g(m_1, m_2, \dots, m_n) = m_1 \text{xor } m_2 \text{ xor } \dots \text{xor } m_n$

Người đi trước thắng nếu $t \neq 0$, nước đi chiến thắng xác định như sau:

- Nếu bốc ở đống i, số sỏi còn lại m_i' của đống này phải thỏa: t xor m_i xor $m_i' = 0$
- Do $g(x) = x\ \forall x$, số sỏi còn lại của đống i phải là $m_i' = t$ xor m_i
- Vậy nếu $m_i' < m_i$ thì có cách bốc chiến thắng từ đống i.

Ví dụ 8. Staircase-Nim

Trò chơi diễn ra trên một cầu thang N bậc đánh số $1,2,\ldots,N$, chân cầu thang được coi là bậc 0. Tại bậc i có x_i viên sỏi $\forall i > 0$. Hai người chơi luân phiên thực hiện lượt chơi, ở mỗi lượt người chơi chuyển một số lượng nguyên dương sỏi ở một bậc nào đó xuống bậc liền dưới. Người không thực hiện được lượt chơi của mình là người thua cuộc.

Phân tích

- Khả năng chiến thắng chỉ phụ thuộc vào số sỏi ở bậc lẻ, vì nếu người chơi trước thực hiện chuyển k sỏi từ bậc $2p+2$ xuống bậc $2p+1$, người chơi sau chỉ cần chuyển k sỏi đó xuống bậc $2p$ và dãy số lượng sỏi ở bậc lẻ vẫn giữ nguyên.
- Như vậy có thể bỏ qua số lượng sỏi ở các bậc chẵn, và coi mỗi lượt chơi chỉ là chuyển sỏi từ các bậc lẻ. Khi đó mỗi lượt chơi tương đương với việc bốc bỏ sỏi ở một bậc lẻ.
- Trò chơi tương đương với trò chơi Nim với các đống sỏi ở bậc lẻ.

Ví dụ 9. Bốc sỏi không giảm

Trò chơi diễn ra với N đống sỏi đánh số $1,2,\ldots,N$, số sỏi của đống i là x_i, dãy $(x_1,x_2,\ldots,x_N)$ phải được đảm bảo là dãy không giảm trong mọi thời điểm. Hai người chơi luân phiên thực hiện lượt chơi, ở mỗi lượt, người chơi phải bốc bỏ khỏi một đống nào đó một lượng sỏi nguyên dương (và vẫn đảm bảo dãy số lượng sau đó là dãy không giảm). Người không thực hiện được lượt chơi của mình là người thua cuộc.

Phân tích

Xét dãy số là chênh lệch sỏi của các đống liên tiếp: $y_1 = x_1, y_2 = x_2 - x_1, \ldots, y_N = x_N - x_{N-1}$. Dãy này sẽ gồm toàn số nguyên không âm trong suốt tiến trình chơi.

Việc bốc số lượng sỏi k khỏi đống i sẽ làm giảm y_i và tăng y_{i+1} một lượng k. Nghĩa là trò chơi tương đương với Staircase-Nim có dãy sỏi ở các bậc $1,2,\ldots,N$ tương ứng là $y_N, y_{N-1},\ldots,y_1$.

Ví dụ 10. Misère Nim

Trò chơi Nim với quy định ai phải bốc viên sỏi cuối cùng là người thua cuộc.

Phân tích

- Nếu tất cả các đống sỏi đều có đúng 1 viên sỏi thì người đi trước (An) thắng khi và chỉ khi số đống sỏi là một số chẵn.
- Nếu chỉ có đúng một đống sỏi i có số sỏi lớn hơn 1 thì An tất thắng vì có thể điều khiển được số lượng đống sỏi có đúng 1 sỏi bằng cách bốc toàn bộ hay chừa lại 1 sỏi ở đống i.
- Trong trò chơi Nim, nước đi chiến thắng không bao giờ đưa đến trạng thái chỉ còn một đống sỏi.
- Vậy An sẽ chơi theo quy luật của trò chơi Nim cho đến khi nhận được trạng thái chỉ còn đúng một đống sỏi có số lượng sỏi lớn hơn 1, An sẽ bốc tất cả hoặc chừa lại 1 sỏi trong đống này như phân tích ở trên.

Ví dụ 11. Pawns (ROI)

Một trò đối kháng diễn ra giữa hai người được mô tả một cách đơn giản như sau: Có N ô trên bàn cờ đặc biệt, với mỗi ô có một tập các ô khác mà nó có thể đi đến sau một nước đi, sao cho thoả mãn rằng không tồn tại một dãy các nước đi xuất phát từ một điểm sau hữu hạn bước thì quay trở lại ô ban đầu. Hiện đang có T quân cờ ở T ô có thể trùng nhau. Hai người luân phiên nhau đi, tại mỗi một bước người chơi có thể cầm một quân cờ di chuyển đến một ô mà nó có thể đi đến sau một bước đi. Trò chơi kết thúc khi không thể đi tiếp. Người không thể đi được là người thua cuộc. Cho biết trước trạng thái ban đầu, và tập các ô có thể đi đến ở các đỉnh. Yêu cầu xác định người đi trước thắng hay thua?

Input

- Dòng đầu ghi số N (1<=N<=500) và M (1<=M<=5000) là số ô và số đường đi trực tiếp nối các đỉnh của bàn cờ.
- M dòng tiếp mỗi dòng ghi cặp số x y thể hiện cạnh đi từ ô có số hiệu x sang ô có số hiệu y.
- Dòng tiếp theo ghi số T là số lượng test.
- T dòng tiếp theo mỗi dòng gồm N số thể hiện số lượng quân tốt ở các ô trong mỗi test.

Ouput

- Tương ứng với mỗi test ghi số 1 hoặc thể hiện người đi đầu thắng hay thua.

Ví dụ

PAWNS.IN	PAWNS.OUT
4 4 2 1 2 3 3 4 2 0 2 0 0 0 1 1 0	0 1

Phân tích

- Đồ thị biểu diễn đi là đồ thị định hướng và không có chu trình, suy ra thoả mãn điều kiện có thể áp dụng thuật toán dùng hàm grundy.
- Trước tiên tính các giá trị của hàm grundy, sau đó áp dụng công thức cộng đồ thị để suy ra kết cục thắng thua.

Chương trình

```
Các mảng:
grundy : array [0…500] of Integer;
Exist, Visit: array [0…500] of byte;
//*Exist[i] = 1 thì đỉnh đó có đi đến đỉnh có hàm grundy bằng 1
//*Visit[x] = 1 thì đã tính hàm grundy tại x rồi.

Procedure Tinh_ham_grundy(x: integer);
Var y: Integer;
Begin
   Visit[x] := 1;
   Fillchar (Exist, sizeof (Exist), 0);
   For y := 1 to N do
      If Co_duong_noi (x, y) then
      Begin
         If (Visit[y] = 0) then Tinh_ham_grundy(y);
```

```
         Exist[grundy[y]] := 1;
      End;
   y:= 0;
   while (grundy[y] = 1) do inc (y);
   grundy[x] = y;
End;

Procedure Main;
Var x: integer;
Begin
   Fillchar (Visit, sizeof (Visit), 0);
   For x := 1 to N do
      If Visit[x] = 0 Then Tinh_ham_grundy(i);
End;
```

Bài trên ta chỉ dùng cộng đồ thị ở phần tính kết quả, còn bài dưới đây sẽ dùng công thức cộng đồ thị ở phần quy hoạch động.

Ví dụ 12. PAS

Cho mảng kích thước $1 \times M$, có 3 loại thanh mầu trắng, đỏ, xanh kích thước $1 \times w$, $1 \times r$, và $1 \times b$. Một trò chơi được diễn ra giữa hai người, tại mỗi một bước đi thì người chơi có dùng một trong ba loại thanh màu dán lên bảng $1 \times M$, sao cho không đè lên thanh đã dán trước đó. Trò chơi kết thúc khi không thể đi tiếp được nữa, người không đi được sẽ thua. Yêu cầu xác định tính thắng thua của người đi đầu.

Input

- Dòng đầu ghi ba số w, r, b
- Dòng thức hai ghi số m - là số lượng ván chơi
- m dòng tiếp theo mỗi dòng ghi hai số mỗi dòng ghi một số M_i

Giới hạn

- $1 \le w, r, b \le 1000, 1 \le M_i, m \le 1000$

Output

- Tương ứng với một ván chơi ghi số 1 nếu người đi đầu thắng, hoặc 2 nếu thua.

Ví dụ

PAS.IN	PAS.OUT
1 5 1 3 1 5 6	1 1 2

Phân tích

- Nếu coi một bảng kích thước $1 \times m$ là đỉnh của đồ thị thì mỗi trạng thái đi gồm một hay nhiều đỉnh như vậy.
- Đồ thị này cũng là đồ thị định hướng và không chu trình nên áp dụng được hàm grundy.

- Khi đi từ một trạng thái là một kích thước $1 \times m$ bằng cách dán thêm một thanh màu vào thì đồ thị chia làm hai phần. Hai phần này coi như hai thành phần độc lập (có bước đi không ảnh hưởng đến nhau) suy ra ta có thể coi nó là một đồ thị gồm hai đồ thị thành phần riêng biệt có giá hàm grundy tính theo công thức cộng đồ thị ở trên
- Tức là từ bảng $1 \times m$ sau một lần dán thêm thanh màu vào sẽ tách ra thành hai phần là $1 \times x$ và $1 \times y$ thì ta sẽ coi đỉnh $1 \times m$ của đồ thị đến đỉnh có hàm grundy bằng $Grundy[x]\ xor\ Grundy[y]$.

Chương trình

```
Procedure Tinh_ham_GRUNDY(m : integer);
Var
Begin
   Fillchar (Exist, sizeof (Exist), 0);
   For i := 1 to m do
      For j := 1 to 3 do
         If (i >= d[j]) then Exist[grundy[m-i] xor grundy[i-d[j]]] := 1;
   i := 0;
   while (Exist[i] = 1) do inc (i);
   grundy[m] := i;
End;
```

Các quy tắc cần nhớ khi áp dụng hàm grundy

1. Đồ thị tính hàm grundy là đồ thị định hướng không chu trình, nếu có nhiều thành phần thì các thành phần phải độc lập.
2. Công thức xác định hàm grundy: $S(v) = \{g(u)|u \in E(v)\}$; $g(u) = min\{p|(p \in N) \cap (p \notin S(u))\}$
3. Công thức cộng đồ thị: $g(v_1, v_2, \dots, v_n) = g_1(v_1)\ XOR\ g_2(v_2)\ XOR \dots XOR\ g_n(v_n)$.
4. $g(x) = 0 \Leftrightarrow$ trạng thái x ở thế thua.

[to be continued]