



# Wormholes

**Link submit:**

[https://uva.onlinejudge.org/index.php?option=com\\_onlinejudge&Itemid=8&page=show\\_problem&problem=499](https://uva.onlinejudge.org/index.php?option=com_onlinejudge&Itemid=8&page=show_problem&problem=499)

**Solution:**

C++	<a href="http://ideone.com/ZlcugR">http://ideone.com/ZlcugR</a>
Java	<a href="https://ideone.com/QF5KxM">https://ideone.com/QF5KxM</a>
Python	<a href="https://ideone.com/88iliR">https://ideone.com/88iliR</a>

**Tóm tắt đề:**

Wormholes (lỗ sâu vũ trụ) là một đường đi xuyên qua không gian và thời gian kết nối hai điểm khác nhau (hai điểm có thể thuộc hai hệ sao khác nhau star systems), về mặt lý thuyết có thể giảm thời gian và quãng đường di chuyển.

Các nhà khoa học cho rằng việc đi qua đi lại nhiều lần một chu trình trên một wormholes có thể giúp họ trở lại thời điểm trong quá khứ bất cứ khi nào, họ có thể trở về và nhìn thấy vụ nổ Big Bang bằng mắt thường.

Hãy xác định xem trên các wormholes có tồn tại chu trình lặp lại mãi mãi hay không. Nghĩa là bạn tìm một chu trình âm trên đồ thị hay không? Nếu có thì in ra "possible", ngược lại thì in ra "not possible".

**Input:**

Dòng đầu tiên chứa số lượng bộ test. Mỗi bộ test gồm các thông tin:

Dòng đầu chứa hai số nguyên  $n$   $m$  ( $1 \leq n \leq 1.000$ ,  $0 \leq m \leq 2.000$ ) lần lượt là số lượng các hệ sao (star systems) và số lượng các wormholes. Star systems được đánh số từ 0 đến  $n - 1$ , trong đó 0 là hệ mặt trời.

$m$  dòng tiếp theo mỗi dòng chứa thông tin của một wormholes gồm ba số nguyên  $x$ ,  $y$ ,  $t$  đại diện cho hệ sao  $x$  kết nối với hệ sao  $y$  và số  $t$  ( $-1.000 \leq t \leq 1.000$ ) là thời gian di chuyển trong wormhole đó.

**Output:**

Có  $c$  dòng mỗi dòng đại diện cho mỗi testcase. Nếu tồn tại chu trình âm thì in ra "possible", ngược lại in ra "not possible".

**Ví dụ:**

2	possible
3 3	not possible
0 1 1000	
1 2 15	
2 1 -42	
4 4	
0 1 10	
1 2 20	
2 3 30	
3 0 -60	

**Giải thích ví dụ:**

Ví dụ gồm hai bộ test:

*Bộ 1:* tồn tại một chu trình âm, do đi từ 1 → 2 là 15, nhưng đi ngược lại 2 → 1 chi phí là -42. Vì thế in ra là “possible”.

*Bộ 2:* không tồn tại chu trình âm nào hết nên in ra là “not possible”.

**Hướng dẫn giải:**

Trong thuật toán Bellman-Ford có vòng lặp ở phía dưới cùng, nó tương tự hai vòng lặp phía trên dùng để cập nhật đường đi. Vòng lặp phía dưới đó là thao tác kiểm tra chu trình âm, giúp phát hiện trong đồ thị có chu trình nào có giá trị âm hay không.

**Độ phức tạp:**  $O(T * V * E)$  với T là số lượng bộ test, V là số lượng đỉnh và E là số lượng cạnh của đồ thị.