



## 558 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++:** <http://ideone.com/ZlcugR>

**python:** <https://ideone.com/88iliR>

**Java:** <https://ideone.com/QF5KxM>

**Tóm tắt đề:** wormholes (Lỗ sâu vũ trụ/lỗ giun vũ trụ) là một đường đi xuyên qua không gian và thời gian kết nối hai điểm khác nhau (2 điểm có thể thuộc 2 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 2 số  $n$  ( $1 \leq n \leq 1000$ ) là số lượng các hệ sao (star systems), số tiếp theo là  $m$  ( $0 \leq m \leq 2000$ ) là số lượng các wormholes.

$m$  dòng tiếp theo mỗi dòng chứa thông tin của một wormholes gồm 3 số:  $x, y, t$ .

Với  $x, y$  là đại diện cho hệ sao  $x$  kết nối với hệ sao  $y$  và số  $t$  ( $-1000 \leq t \leq 1000$ ) 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".

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:** Ở ví dụ đầu tiên tồn tại một chu trình âm, do đi từ 1 → 2 là 15, nhưng đi ngược lại 2 → 1 chỉ phí là -42. Vì thế in ra là “possible”.

Ở ví dụ thứ 2 thì không có tồn tại chung trình âm nào hết nên in ra là “not possible”

**Hướng dẫn:** Thuật toán Bellman-Ford có thêm dòng for phía dưới, tương tự 2 dòng for phía trên dùng để cập nhật đường đi. 1 dòng for phía dưới sẽ phát hiện ra 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.