Mummy Maze

Hiện tại, bài tập này đã có trên online judge chính thức của VNOI, bạn có thể truy cập ở đây: https://oj.vnoi.info/problem/vmmummy

Nếu đã từng có một thời chơi những trò chơi đơn giản mà thú vị của Popcap Game, chắc hẳn bạn không thể không biết đến trò chơi Mummy Maze. Trong trò chơi này, bạn vào vai một Nhà thám hiểm với nhiệm vụ khám phá các Kim Tự Tháp Ai Cập, và tìm kiếm những kho tàng vô giá được để lại từ hàng nghìn năm trước. Dĩ nhiên, với phần thưởng lớn như vậy, chuyến đi của bạn sẽ gặp vô vàn gian nan trắc trở do có rất nhiều cạm bẫy, bọ cạp, và đáng sợ hơn cả là xác ướp đang chờ bạn ở phía trước...

Trong mỗi ván chơi, bạn cần tìm đường thoát khỏi một mê cung trong Kim Tự Tháp mà không bị bắt bởi **M** xác ướp (xác ướp bắt được bạn nếu bạn và Xác ướp đứng ở cùng một vị trí). Mê cung có dạng một hình vuông kích thước **N** x **N**, được chia thành các ô vuông đơn vị 1 x 1. Ô nằm ở hàng i, cột j được gọi là ô (i, j). Giữa các ô vuông trong mê cung có một số bức tường ngăn không cho bạn di chuyển giữa một số ô. Để đơn giản, chúng ta sẽ đánh số các hàng và cột phía bên trong của mê cung bằng các số nguyên từ 1 đến N. Hàng nằm ngay phía trên hàng 1 (và nằm ngoài mê cung) được đánh số 0, hàng nằm ngay phía dưới hàng N (và nằm ngoài mê cung) được đánh số N + 1. Tương tự với cột 0 và cột N + 1. Bạn xuất phát từ ô (**sx**, **sy**) và cần di chuyển đến ô (**fx**, **fy**). Ô (sx, sy) luôn nằm trong mê cung và ô (fx, fy) luôn nằm ngoài và kề cạnh với một ô trong mê cung (nghĩa là fx = 0 hoặc fx = N + 1 hoặc fy = 0 hoặc fy = N + 1).

Trò chơi được thực hiện theo lượt, bạn và M xác ướp lần lượt di chuyển. Bạn được quyền di chuyển trước. Sau khi bạn di chuyển, M xác ướp sẽ đồng thời di chuyển, rồi sau đó lại đến lượt bạn di chuyển. Trò chơi tiếp tục đến khi bạn thoát khỏi mê cung hoặc bị xác ướp bắt được. Bạn và xác ướp di chuyển theo nguyên tắc sau:

- Ở lượt bạn di chuyển, bạn được quyền **đứng im hoặc di chuyển sang 4 ô kề cạnh** với ô hiện tại của bạn.
- Trong lượt đi của mình, các xác ướp cũng di chuyển theo cách tương tự, nghĩa là với mỗi bước xác ướp có quyền đứng im hoặc di chuyển sang 4 ô kề cạnh. Nhưng đặc biệt, chúng được quyền đi **tối đa 2 bước trong một lượt**. Mỗi bước di chuyển của xác ướp đều nhằm làm khoảng cách Manhattan giữa bạn và chúng nhỏ nhất có thể. Tuy nhiên, có hai loại xác ướp với hai cách di chuyển khác nhau:
  - Loại xác ướp màu đỏ (đánh số 0) luôn **ưu tiên di chuyển dọc** (lên/xuống). Điều này có nghĩa là, nếu trong một lần di chuyển của xác ướp đỏ, mà việc di chuyển dọc hoặc ngang (trái/phải) đều khiến xác ướp đến gần bạn hơn, thì xác ướp đỏ sẽ chọn di chuyển dọc. Nếu việc di chuyển dọc không làm xác ướp tiến tới gần bạn hơn, hoặc xác ướp không thể di chuyển dọc, xác ướp mới di chuyển ngang.
  - Loại xác ướp màu trắng (đánh số 1) thì ngược lại, luôn **ưu tiên di chuyển ngang**.
- Tại một thời điểm, một ô có thể có nhiều hơn một xác ướp.
- Cả bạn và xác ướp đều không thể di chuyển xuyên qua tường. Bạn chỉ có thể rời khỏi mê cung nếu ô đích là ô (fx, fy). Các xác ướp chỉ được di chuyển trong mê cung. Do đó, một khi bạn đến được ô (fx, fy), bạn sẽ giành chiến thắng mà không cần quan tâm đến lượt di chuyển của xác ướp ngay sau đó vì chúng không được di chuyển ra ngoài mê cung.

- Vị trí ban đầu của các xác ướp luôn nằm trong bảng và không trùng với ô (sx, sy).

Chú ý: *Trong phần mô tả trò chơi ở bài này, một số chi tiết của trò chơi được bỏ qua hoặc thay đổi, vì vậy nếu bạn đã từng chơi trò chơi này rồi, bạn vẫn nên đọc qua phần mô tả trò chơi. Nếu vẫn chưa hiểu rõ cách thức di chuyển của bạn và xác ướp, bạn có thể xem thêm đoạn clip tại đây hoặc download trò chơi Mummy Maze về chơi thử.*

Vì xác ướp có thể di chuyển nhanh gấp 2 lần bạn, nên việc thoát khỏi mê cung đòi hỏi bạn phải sử dụng khôn khéo các bức tường để làm chậm xác ướp lại. Trong khi mọi người đang vắt tay lên trán để suy nghĩ nước đi, bạn, một lập trình viên siêu đẳng, chỉ cần viết một chương trình để tính toán nước đi cho mình.

Để thuận tiện cho việc lập trình, bạn đánh số các cách di chuyển của mình bằng các chữ số 0, 1, 2, 3 như hình dưới. Đứng im được đánh số 4. Một cách di chuyển sẽ được biểu diễn bằng một số, trong đó mỗi chữ số thể hiện bước đi tương ứng. Ví dụ, cách di chuyển: đi lên, đi sang trái và đi xuống sẽ được biểu diễn bằng số 320.

Vì việc qua level quá đơn giản nên bạn quyết định thử thách mình thêm một chút. Nhiệm vụ mới mà bạn đặt ra là **tìm cách di chuyển để thoát khỏi mê cung trong thời gian ngắn nhất** (sử dụng ít lượt nhất). Nếu có nhiều phương án, bạn cần đưa ra cách di chuyển có số biểu diễn là **nhỏ nhất khi đọc từ phải sang trái**. Ví dụ nếu bạn tìm được 2 cách di chuyển là 13320 và 23310 thì kết quả sẽ là 23310 vì 01332 < 02331.

Hình minh họa cho test ví dụ đầu tiên

Input

Trong mỗi file input, bạn được cho một mô tả của một mê cung, theo định dạng dưới đây:

- Dòng đầu tiên là số nguyên dương **N** - kích thước của mê cung.
- Dòng thứ hai chứa 4 số nguyên **sx sy fx fy**, cách nhau bởi ít nhất một dấu cách.
- Dòng thứ ba là số nguyên dương **M** - số lượng xác ướp trong mê cung.
- M dòng tiếp theo mô tả các xác ướp. Dòng thứ i gồm 3 số nguyên $\mathbf{x_i\ y_i\ z_i}$, trong đó (x_i, y_i) là ô ban đầu của xác ướp, và z_i là loại xác ướp ($z_i = 0$ hoặc 1).
- Dòng tiếp theo là số nguyên **WH** - số bức tường ngang.
- Tiếp theo là WH dòng, dòng thứ i là cặp số nguyên $\mathbf{u_i\ v_i}$, cho biết có một bức tường ngăn cách giữa ô (u_i, v_i) và ô $(u_i + 1, v_i)$. Ô (u_i, v_i) và ô $(u_i + 1, v_i)$ luôn nằm trong mê cung.
- Dòng tiếp theo là số nguyên **WV** - số bức tường dọc.
- Tiếp theo là WV dòng, dòng thứ i là cặp số nguyên $\mathbf{u_i\ v_i}$, cho biết có một bức tường ngăn cách giữa ô (u_i, v_i) và ô $(u_i, v_i + 1)$. Ô (u_i, v_i) và ô $(u_i, v_i + 1)$ luôn nằm trong mê cung.

Output

Xuất ra số biểu diễn cách di chuyển theo yêu cầu ở trên.

Giới hạn

Trong tất cả các test:

- $1 \le N \le 10$
- $0 \le M \le 2$

Trong 30% số test M = 0.

Trong 30% số test tiếp theo M = 1.

Trong những test còn lại M = 2.

Dữ liệu đảm bảo luôn có cách di chuyển để thoát khỏi mê cung.

Example

Input:

```
6
3 5 7 2
1
6 6 1
4
4 1
4 4
5 4
5 5
7
1 3
3 2
3 3
4 3
```

5 3
5 5
6 4

Output:
322200000

Input:
5
4 4 0 2
2
2 1 0
3 5 0
4
1 2
1 3
3 2
4 4
4
4 2
4 4
5 2
5 3

Output:
2033313223