

Sheep

Big-O Blue - Lecture 5: BFS

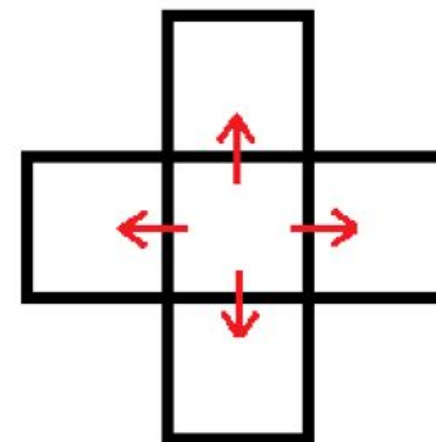
Tóm tắt đề bài

Tóm tắt đề bài

- Có một mảnh đất có dạng một hình chữ nhật, chứa các ô như sau:
 - Kí tự '.' : ô trống đi vào được.
 - Kí tự '#' : hàng rào, không đi vào được.
 - Kí tự 'k' : một chú cừu.
 - Kí tự 'v' : một con sói.

1. Cách di chuyển:

Đứng tại một ô bất kỳ, ta có thể di chuyển sang các ô kề cạnh theo 4 hướng, dĩ nhiên không được đi ra khỏi bảng và không đi vào ô '#'.



Tóm tắt đề bài

2. Khu vực (sector)

Hai ô tồn tại đường đi tới nhau được gọi là chung một **khu vực**.

- Xét trong một khu vực:
 - Số con cừu **lớn hơn hẳn** số con sói **trong cùng một khu vực**, đàn sói sẽ chết hết và đàn cừu sẽ bảo toàn lực lượng.
 - Ngược lại thì đàn cừu = 0, và đàn sói bảo toàn lực lượng.

.	#	#	#	#	#	#	.
#	.	.	k	.	.	.	#
#	.	#	#	#	#	.	#
#	.	#	v	.	#	.	#
#	.	#	.	k	#	k	#
#	k	.	#	#	.	.	#
#	.	v	.	.	v	.	#
.	#	#	#	#	#	#	.

Tóm tắt đề bài

- Nếu khu vực nằm bên rìa mảnh đất thì đàn cừu và sói trong khu vực này đều sống sót.

Yêu cầu: Xác định số cừu và sói còn lại sau một đêm.

.	#	#	#	#	#	#	.
#	.	.	k	.	.	.	#
#	.	#	#	#	#	.	#
#	.	#	v	.	#	.	#
#	.	#	.	k	#	k	#
#	k	.	#	#	.	.	#
#	.	v	.	.	v	.	#
.	#	#	#	#	#	#	.

Mô tả Input/Output

Input:

N M - số lượng hàng, cột của mảnh đất
N dòng tiếp theo chứa M kí tự đại diện
cho mỗi ô của mảnh đất.

$3 \leq N, M \leq 250$

Output:

Số lượng cừu và sói sống sót.

Giải thích ví dụ

Test ví dụ

$N = 8, M = 8$

.	#	#	#	#	#	#	.
#	.	.	k	.	.	.	#
#	.	#	#	#	#	.	#
#	.	#	v	.	#	.	#
#	.	#	.	k	#	k	#
#	k	.	#	#	.	.	#
#	.	v	.	.	v	.	#
.	#	#	#	#	#	#	.

Test ví dụ

N = 8, M = 8

.	#	#	#	#	#	#	.
#	.	.	k	.	.	.	#
#	.	#	#	#	#	.	#
#	.	#	v	.	#	.	#
#	.	#	.	k	#	k	#
#	k	.	#	#	.	.	#
#	.	v	.	.	v	.	#
.	#	#	#	#	#	#	.

□ **Output: 3 1**

	Các khu vực nằm bên rìa mảnh đất.
	Khu vực 1 (sói = 2, cừu = 3) □ Đàn sói chết hết, đàn cừu sống sót đầy đủ. □ Còn 3 cừu
	Khu vực 2 (sói = cừu = 1) □ Cừu chết, còn một mình con sói no bụng □ Còn 1 sói

Hướng dẫn giải

Nhận xét

- Ta xem các ô trên mảnh đất là các đỉnh của đồ thị.
- Giữa hai đỉnh có cạnh nối trực tiếp khi hai ô trên mảnh đất kề cạnh nhau.

	$(x - 1, y)$	
$(x, y - 1)$	(x, y)	$(x, y + 1)$
	$(x + 1, y)$	

$$(x, y - 1) \longrightarrow (x, y) + (0, -1)$$

$$(x, y + 1) \longrightarrow (x, y) + (0, 1)$$

$$(x - 1, y) \longrightarrow (x, y) + (-1, 0)$$

$$(x + 1, y) \longrightarrow (x, y) + (1, 0)$$

$$dr = [0, 0, -1, 1]$$

$$dc = [-1, 1, 0, 0]$$

Nhận xét

- Khi đó, ta chỉ cần duyệt BFS bắt đầu từ một ô không phải là hàng rào để đi đến các ô khác cùng khu vực với nó.
- Đồng thời cũng phải lưu lại số cừu và số sói trong lúc duyệt BFS.
- Nếu khu vực đang xét nằm bên rìa mảnh đất:
→ Số sói và cừu vừa đếm được bảo toàn.
- Ngược lại thì so sánh số cừu và số sói trong khu vực vừa duyệt và cập nhật đáp án.

Nhận xét

Khu vực bên rìa mảnh đất: chứa ≥ 1 ô nằm trên rìa mảnh đất.

.	.	#	#	#	#	#	.
#	.	.	k	.	.	.	#
#	.	#	#	#	#	.	#
#	.	#	v	.	#	.	#
#	.	#	.	k	#	k	#
#	k	.	#	#	.	.	#
#	.	v	.	.	v	.	#
.	#	#	#	#	#	#	.

Ô (x,y) nằm trên rìa mảnh đất khi một trong các điều kiện sau thỏa:

- $x == 0$
- $x == N-1$
- $y == 0$
- $y == M-1$

với $0 \leq x < N$, $0 \leq y < M$

Ý tưởng

.	#	#	#	#	#	#	.
#	.	.	k	.	.	.	#
#	.	#	#	#	#	.	#
#	.	#	v	.	#	.	#
#	.	#	.	k	#	k	#
#	k	.	#	#	.	.	#
#	.	v	.	.	v	.	#
.	#	#	#	#	#	#	.

Khu vực 1: nằm bên rìa mảnh đất

Sheep = 0

Wolf = 0

- Số cừu và sói bảo toàn

Ý tưởng

.	#	#	#	#	#	#	.
#	.	.	k	.	.	.	#
#	.	#	#	#	#	.	#
#	.	#	v	.	#	.	#
#	.	#	.	k	#	k	#
#	k	.	#	#	.	.	#
#	.	v	.	.	v	.	#
.	#	#	#	#	#	#	.

Khu vực 2: nằm bên rìa mảnh đất

Sheep = 0

Wolf = 0

- Số cừu và sói bảo toàn

Ý tưởng

.	#	#	#	#	#	#	.
#	.	.	k	.	.	.	#
#	.	#	#	#	#	.	#
#	.	#	v	.	#	.	#
#	.	#	.	k	#	k	#
#	k	.	#	#	.	.	#
#	.	v	.	.	v	.	#
.	#	#	#	#	#	#	.

Khu vực 2: nằm bên rìa mảnh đất

Sheep = 0

Wolf = 0

- Số cừu và sói bảo toàn

Ý tưởng

.	#	#	#	#	#	#	.
#	.	.	k	.	.	.	#
#	.	#	#	#	#	.	#
#	.	#	v	.	#	.	#
#	.	#	.	k	#	k	#
#	k	.	#	#	.	.	#
#	.	v	.	.	v	.	#
.	#	#	#	#	#	#	.

Khu vực 3

Ý tưởng

.	#	#	#	#	#	#	.
#	.	.	k	.	.	.	#
#	.	#	#	#	#	.	#
#	.	#	v	.	#	.	#
#	.	#	.	k	#	k	#
#	k	.	#	#	.	.	#
#	.	v	.	.	v	.	#
.	#	#	#	#	#	#	.

Khu vực 3: Bên trong mảnh đất

Sheep = 3

Wolf = 2

- Số cừu và sói thay đổi:

Sheep = 3

Wolf = 0

Ý tưởng

.	#	#	#	#	#	#	.
#	.	.	k	.	.	.	#
#	.	#	#	#	#	.	#
#	.	#	v	.	#	.	#
#	.	#	.	k	#	k	#
#	k	.	#	#	.	.	#
#	.	v	.	.	v	.	#
.	#	#	#	#	#	#	.

Khu vực 4

Ý tưởng

.	#	#	#	#	#	#	.
#	.	.	k	.	.	.	#
#	.	#	#	#	#	.	#
#	.	#	v	.	#	.	#
#	.	#	.	k	#	k	#
#	k	.	#	#	.	.	#
#	.	v	.	.	v	.	#
.	#	#	#	#	#	#	.

Khu vực 4: Bên trong mảnh đất

Sheep = 1

Wolf = 1

- Số cừu và sói thay đổi

Sheep = 0

Wolf = 1

Ý tưởng

.	#	#	#	#	#	#	.
#	.	.	k	.	.	.	#
#	.	#	#	#	#	.	#
#	.	#	v	.	#	.	#
#	.	#	.	k	#	k	#
#	k	.	#	#	.	.	#
#	.	v	.	.	v	.	#
.	#	#	#	#	#	#	.

Khu vực 5: nằm bên rìa mảnh đất

Sheep = 0

Wolf = 0

- Số cừu và sói bảo toàn

Ý tưởng

.	#	#	#	#	#	#	.
#	.	.	k	.	.	.	#
#	.	#	#	#	#	.	#
#	.	#	v	.	#	.	#
#	.	#	.	k	#	k	#
#	k	.	#	#	.	.	#
#	.	v	.	.	v	.	#
.	#	#	#	#	#	#	.

Khu vực 6: nằm bên rìa mảnh đất

Sheep = 0

Wolf = 0

- Số cừu và sói bảo toàn

Mô phỏng thuật toán:

- Ta sẽ mô phỏng thuật toán duyệt BFS lên test ví dụ:

.	#	#	#	#	#	#	.
#	.	.	k	.	.	.	#
#	.	#	#	#	#	.	#
#	.	#	v	.	#	.	#
#	.	#	.	k	#	k	#
#	k	.	#	#	.	.	#
#	.	v	.	.	v	.	#
.	#	#	#	#	#	#	.

Queue:

Sheep: 0

Wolf: 0

ConnectedOutside: **False**

Mô phỏng thuật toán:

.	#	#	#	#	#	#	.
#	.	.	k	.	.	.	#
#	.	#	#	#	#	.	#
#	.	#	v	.	#	.	#
#	.	#	.	k	#	k	#
#	k	.	#	#	.	.	#
#	.	v	.	.	v	.	#
.	#	#	#	#	#	#	.

Queue: (1, 1)

Sheep: 0

Wolf: 0

ConnectedOutside: False

Mô phỏng thuật toán:

.	#	#	#	#	#	#	.
#	.	.	k	.	.	.	#
#	.	#	#	#	#	.	#
#	.	#	v	.	#	.	#
#	.	#	.	k	#	k	#
#	k	.	#	#	.	.	#
#	.	v	.	.	v	.	#
.	#	#	#	#	#	#	.

Queue:

Sheep: 0

Wolf: 0

ConnectedOutside: **True**

Queue rỗng ☐ BFS kết thúc.

Mô phỏng thuật toán:

.	#	#	#	#	#	#	.
#	.	.	k	.	.	.	#
#	.	#	#	#	#	.	#
#	.	#	v	.	#	.	#
#	.	#	.	k	#	k	#
#	k	.	#	#	.	.	#
#	.	v	.	.	v	.	#
.	#	#	#	#	#	#	.

Queue: (1, 8)

Sheep: 0

Wolf: 0

ConnectedOutside: **False**

Mô phỏng thuật toán:

.	#	#	#	#	#	#	.
#	.	.	k	.	.	.	#
#	.	#	#	#	#	.	#
#	.	#	v	.	#	.	#
#	.	#	.	k	#	k	#
#	k	.	#	#	.	.	#
#	.	v	.	.	v	.	#
.	#	#	#	#	#	#	.

Queue:

Sheep: 0

Wolf: 0

ConnectedOutside: **True**

Queue rỗng ☐ BFS kết thúc.

Mô phỏng thuật toán:

.	#	#	#	#	#	#	.
#	.	.	k	.	.	.	#
#	.	#	#	#	#	.	#
#	.	#	v	.	#	.	#
#	.	#	.	k	#	k	#
#	k	.	#	#	.	.	#
#	.	v	.	.	v	.	#
.	#	#	#	#	#	#	.

Queue: (2, 2)

Sheep: 0

Wolf: 0

ConnectedOutside: **False**

Mô phỏng thuật toán:

.	#	#	#	#	#	#	.
#	.	.	k	.	.	.	#
#	.	#	#	#	#	.	#
#	.	#	v	.	#	.	#
#	.	#	.	k	#	k	#
#	k	.	#	#	.	.	#
#	.	v	.	.	v	.	#
.	#	#	#	#	#	#	.

Queue: (2, 3), (3, 2)

Sheep: 0

Wolf: 0

ConnectedOutside: **False**

Mô phỏng thuật toán:

.	#	#	#	#	#	#	.
#	.	.	k	.	.	.	#
#	.	#	#	#	#	.	#
#	.	#	v	.	#	.	#
#	.	#	.	k	#	k	#
#	k	.	#	#	.	.	#
#	.	v	.	.	v	.	#
.	#	#	#	#	#	#	.

Queue: (3, 2), (2, 4)

Sheep: 0

Wolf: 0

ConnectedOutside: False

Mô phỏng thuật toán:

.	#	#	#	#	#	#	.
#	.	.	k	.	.	.	#
#	.	#	#	#	#	.	#
#	.	#	v	.	#	.	#
#	.	#	.	k	#	k	#
#	k	.	#	#	.	.	#
#	.	v	.	.	v	.	#
.	#	#	#	#	#	#	.

Queue: (2, 4), (4, 2)

Sheep: 0

Wolf: 0

ConnectedOutside: False

Mô phỏng thuật toán:

.	#	#	#	#	#	#	.
#	.	.	k	.	.	.	#
#	.	#	#	#	#	.	#
#	.	#	v	.	#	.	#
#	.	#	.	k	#	k	#
#	k	.	#	#	.	.	#
#	.	v	.	.	v	.	#
.	#	#	#	#	#	#	.

Queue: (4, 2), (2, 5)

Sheep: 1

Wolf: 0

ConnectedOutside: False

Mô phỏng thuật toán:

.	#	#	#	#	#	#	.
#	.	.	k	.	.	.	#
#	.	#	#	#	#	.	#
#	.	#	v	.	#	.	#
#	.	#	.	k	#	k	#
#	k	.	#	#	.	.	#
#	.	v	.	.	v	.	#
.	#	#	#	#	#	#	.

Queue: (2, 5), (5, 2)

Sheep: 1

Wolf: 0

ConnectedOutside: False

Mô phỏng thuật toán:

.	#	#	#	#	#	#	.
#	.	.	k	.	.	.	#
#	.	#	#	#	#	.	#
#	.	#	v	.	#	.	#
#	.	#	.	k	#	k	#
#	k	.	#	#	.	.	#
#	.	v	.	.	v	.	#
.	#	#	#	#	#	#	.

Queue: (5, 2), (2, 6)

Sheep: 1

Wolf: 0

ConnectedOutside: False

Mô phỏng thuật toán:

.	#	#	#	#	#	#	.
#	.	.	k	.	.	.	#
#	.	#	#	#	#	.	#
#	.	#	v	.	#	.	#
#	.	#	.	k	#	k	#
#	k	.	#	#	.	.	#
#	.	v	.	.	v	.	#
.	#	#	#	#	#	#	.

Queue: (2, 6), (6, 2)

Sheep: 1

Wolf: 0

ConnectedOutside: False

Mô phỏng thuật toán:

.	#	#	#	#	#	#	.
#	.	.	k	.	.	.	#
#	.	#	#	#	#	.	#
#	.	#	v	.	#	.	#
#	.	#	.	k	#	k	#
#	k	.	#	#	.	.	#
#	.	v	.	.	v	.	#
.	#	#	#	#	#	#	.

Queue: (6, 2), (2, 7)

Sheep: 1

Wolf: 0

ConnectedOutside: False

Mô phỏng thuật toán:

.	#	#	#	#	#	#	.
#	.	.	k	.	.	.	#
#	.	#	#	#	#	.	#
#	.	#	v	.	#	.	#
#	.	#	.	k	#	k	#
#	k	.	#	#	.	.	#
#	.	v	.	.	v	.	#
.	#	#	#	#	#	#	.

Queue: (2, 7), (6, 3), (7, 2)

Sheep: 2

Wolf: 0

ConnectedOutside: False

Mô phỏng thuật toán:

.	#	#	#	#	#	#	.
#	.	.	k	.	.	.	#
#	.	#	#	#	#	.	#
#	.	#	v	.	#	.	#
#	.	#	.	k	#	k	#
#	k	.	#	#	.	.	#
#	.	v	.	.	v	.	#
.	#	#	#	#	#	#	.

Queue: (6, 3), (7, 2), (3, 7)

Sheep: 2

Wolf: 0

ConnectedOutside: False

Mô phỏng thuật toán:

.	#	#	#	#	#	#	.
#	.	.	k	.	.	.	#
#	.	#	#	#	#	.	#
#	.	#	v	.	#	.	#
#	.	#	.	k	#	k	#
#	k	.	#	#	.	.	#
#	.	v	.	.	v	.	#
.	#	#	#	#	#	#	.

Queue: (7, 2), (3, 7), (7, 3)

Sheep: 2

Wolf: 0

ConnectedOutside: False

Mô phỏng thuật toán:

.	#	#	#	#	#	#	.
#	.	.	k	.	.	.	#
#	.	#	#	#	#	.	#
#	.	#	v	.	#	.	#
#	.	#	.	k	#	k	#
#	k	.	#	#	.	.	#
#	.	v	.	.	v	.	#
.	#	#	#	#	#	#	.

Queue: (3, 7), (7, 3)

Sheep: 2

Wolf: 0

ConnectedOutside: False

Mô phỏng thuật toán:

.	#	#	#	#	#	#	.
#	.	.	k	.	.	.	#
#	.	#	#	#	#	.	#
#	.	#	v	.	#	.	#
#	.	#	.	k	#	k	#
#	k	.	#	#	.	.	#
#	.	v	.	.	v	.	#
.	#	#	#	#	#	#	.

Queue: (7, 3), (4, 7)

Sheep: 2

Wolf: 0

ConnectedOutside: **False**

Mô phỏng thuật toán:

.	#	#	#	#	#	#	.
#	.	.	k	.	.	.	#
#	.	#	#	#	#	.	#
#	.	#	v	.	#	.	#
#	.	#	.	k	#	k	#
#	k	.	#	#	.	.	#
#	.	v	.	.	v	.	#
.	#	#	#	#	#	#	.

Queue: (4, 7), (7, 4)

Sheep: 2

Wolf: 1

ConnectedOutside: **False**

Mô phỏng thuật toán:

.	#	#	#	#	#	#	.
#	.	.	k	.	.	.	#
#	.	#	#	#	#	.	#
#	.	#	v	.	#	.	#
#	.	#	.	k	#	k	#
#	k	.	#	#	.	.	#
#	.	v	.	.	v	.	#
.	#	#	#	#	#	#	.

Queue: (7, 4), (5, 7)

Sheep: 2

Wolf: 1

ConnectedOutside: False

Mô phỏng thuật toán:

.	#	#	#	#	#	#	.
#	.	.	k	.	.	.	#
#	.	#	#	#	#	.	#
#	.	#	v	.	#	.	#
#	.	#	.	k	#	k	#
#	k	.	#	#	.	.	#
#	.	v	.	.	v	.	#
.	#	#	#	#	#	#	.

Queue: (5, 7), (7, 5)

Sheep: 2

Wolf: 1

ConnectedOutside: False

Mô phỏng thuật toán:

.	#	#	#	#	#	#	.
#	.	.	k	.	.	.	#
#	.	#	#	#	#	.	#
#	.	#	v	.	#	.	#
#	.	#	.	k	#	k	#
#	k	.	#	#	.	.	#
#	.	v	.	.	v	.	#
.	#	#	#	#	#	#	.

Queue: (7, 5), (6, 7)

Sheep: 3

Wolf: 1

ConnectedOutside: False

Mô phỏng thuật toán:

.	#	#	#	#	#	#	.
#	.	.	k	.	.	.	#
#	.	#	#	#	#	.	#
#	.	#	v	.	#	.	#
#	.	#	.	k	#	k	#
#	k	.	#	#	.	.	#
#	.	v	.	.	v	.	#
.	#	#	#	#	#	#	.

Queue: (6, 7), (7, 6)

Sheep: 3

Wolf: 1

ConnectedOutside: False

Mô phỏng thuật toán:

.	#	#	#	#	#	#	.
#	.	.	k	.	.	.	#
#	.	#	#	#	#	.	#
#	.	#	v	.	#	.	#
#	.	#	.	k	#	k	#
#	k	.	#	#	.	.	#
#	.	v	.	.	v	.	#
.	#	#	#	#	#	#	.

Queue: (7, 6), (7, 7), (6, 6)

Sheep: 3

Wolf: 1

ConnectedOutside: **False**

Mô phỏng thuật toán:

.	#	#	#	#	#	#	.
#	.	.	k	.	.	.	#
#	.	#	#	#	#	.	#
#	.	#	v	.	#	.	#
#	.	#	.	k	#	k	#
#	k	.	#	#	.	.	#
#	.	v	.	.	v	.	#
.	#	#	#	#	#	#	.

Queue: (7, 7), (6, 6)

Sheep: 3

Wolf: 2

ConnectedOutside: False

Mô phỏng thuật toán:

.	#	#	#	#	#	#	.
#	.	.	k	.	.	.	#
#	.	#	#	#	#	.	#
#	.	#	v	.	#	.	#
#	.	#	.	k	#	k	#
#	k	.	#	#	.	.	#
#	.	v	.	.	v	.	#
.	#	#	#	#	#	#	.

Queue: (6, 6)

Sheep: 3

Wolf: 2

ConnectedOutside: **False**

Mô phỏng thuật toán:

.	#	#	#	#	#	#	.
#	.	.	k	.	.	.	#
#	.	#	#	#	#	.	#
#	.	#	v	.	#	.	#
#	.	#	.	k	#	k	#
#	k	.	#	#	.	.	#
#	.	v	.	.	v	.	#
.	#	#	#	#	#	#	.

Queue:

Sheep: 3

Wolf: 2

ConnectedOutside: **False**

Đến đây queue rỗng nên BFS dừng lại.

□ Trong khu vực này có 3 cừu và 2 sói

□ Do số cừu > số sói nên cập nhật số cừu sống sót thêm 3

Mô phỏng thuật toán:

.	#	#	#	#	#	#	.
#	.	.	k	.	.	.	#
#	.	#	#	#	#	.	#
#	.	#	v	.	#	.	#
#	.	#	.	k	#	k	#
#	k	.	#	#	.	.	#
#	.	v	.	.	v	.	#
.	#	#	#	#	#	#	.

Queue: (4, 4)

Sheep: 0

Wolf: 0

ConnectedOutside: **False**

Mô phỏng thuật toán:

.	#	#	#	#	#	#	.
#	.	.	k	.	.	.	#
#	.	#	#	#	#	.	#
#	.	#	v	.	#	.	#
#	.	#	.	k	#	k	#
#	k	.	#	#	.	.	#
#	.	v	.	.	v	.	#
.	#	#	#	#	#	#	.

Queue: (4,5), (5, 4)

Sheep: 0

Wolf: 1

ConnectedOutside: **False**

Mô phỏng thuật toán:

.	#	#	#	#	#	#	.
#	.	.	k	.	.	.	#
#	.	#	#	#	#	.	#
#	.	#	v	.	#	.	#
#	.	#	.	k	#	k	#
#	k	.	#	#	.	.	#
#	.	v	.	.	v	.	#
.	#	#	#	#	#	#	.

Queue: (5, 4), (5, 5)

Sheep: 0

Wolf: 1

ConnectedOutside: **False**

Mô phỏng thuật toán:

.	#	#	#	#	#	#	.
#	.	.	k	.	.	.	#
#	.	#	#	#	#	.	#
#	.	#	v	.	#	.	#
#	.	#	.	k	#	k	#
#	k	.	#	#	.	.	#
#	.	v	.	.	v	.	#
.	#	#	#	#	#	#	.

Queue: (5, 5)

Sheep: 0

Wolf: 1

ConnectedOutside: False

Mô phỏng thuật toán:

.	#	#	#	#	#	#	.
#	.	.	k	.	.	.	#
#	.	#	#	#	#	.	#
#	.	#	v	.	#	.	#
#	.	#	.	k	#	k	#
#	k	.	#	#	.	.	#
#	.	v	.	.	v	.	#
.	#	#	#	#	#	#	.

Queue:

Sheep: 1

Wolf: 1

ConnectedOutside: **False**

Queue rỗng □ BFS dừng lại

- Trong vùng vừa duyệt có 1 cừu và 1 sói

□ Cừu chết và cập nhật số sói sống thêm 1.

Mô phỏng thuật toán:

.	#	#	#	#	#	#	.
#	.	.	k	.	.	.	#
#	.	#	#	#	#	.	#
#	.	#	v	.	#	.	#
#	.	#	.	k	#	k	#
#	k	.	#	#	.	.	#
#	.	v	.	.	v	.	#
.	#	#	#	#	#	#	.

Tiếp tục cho hai ô vuông còn lại chưa được duyệt.

□ Mọi ô vuông không phải hàng rào đều được duyệt qua.

□ Xuất ra đáp án của bài toán.

Các bước giải

Bước 1: Đọc vào hai số N, M

Bước 2: Đọc vào bảng hai chiều A đại diện cho khu vườn của đề bài

Bước 3: Khởi tạo các biến đáp án $\text{survivedSheep} = 0, \text{survivedWolf} = 0$

Bước 4: Duyệt i chạy từ 1 đến N , j chạy từ 1 đến M :

 Nếu $A[i][j] \neq \text{'\#'}$ và (i, j) chưa được thăm thì bắt đầu BFS tại (i, j) .

 Cập nhật đáp án sau khi BFS tại (i, j) .

Bước 5: Xuất ra đáp án.

Độ phức tạp: $O(M \times N)$

Mã giả

Mã giả

```
read N, M
a = [None]*N
for i = 0 → N-1:
    read a[i]
visited = [[false]*M for i in range(N)]
nSheep = 0, nWolf = 0
for i = 0 → N-1:
    for j = 0 → M-1:
        if a[i][j] != '#' and visited[i][j] == false:
            bfs(i, j)
print(nSheep, nWolf)
```

Do phuc tap: $O(N*M + N*M) = O(N*M)$

Mã giả

```
dr = [0,0,1,-1]
dc = [1,-1,0,0]
function bfs(i, j):
    q = Queue()
    q.push({i,j})
    visited[i][j] = true
    k = 0, v = 0
    connectedOutside = false
    while q is not empty:
        {x, y} = q.front()
        q.pop()
        if a[x][y] == 'k': k += 1
        if a[x][y] == 'v': v += 1
        if onEdge(x,y): connectedOutside = true
        for id = 0 → 3:
            u = x + dr[id]
            v = y + dc[id]
            if isValid(u,v) and a[u][v] != '#' and visited[u][v] == false:
                q.push({u,v})
                visited[u][v] = true
```

```
#ket thuc while
    if connectedOutside:
        nSheep += k
        nWolf += v
    else:
        if k > v:
            nSheep += k
        else:
            nWolf += v

function onEdge(x, y):
    return x == 0 or x == N-1 or y == 0 or y == M-1

function isValid(x,y):
    return 0 ≤ x < N and 0 ≤ y < M
```

Thank you