

این مسأله را می‌توان به کمک الگوریتم مربوط به مسأله‌ی Maximum Flow حل کرد. به این صورت که به یال‌های گراف ورودی وزن ۱ را می‌دهیم و اگر بیشترین جریان از  $s$  به  $t$  حداقل ۲ بود به این معنی است که حداقل دو مسیر یال‌مجزا در گراف از  $s$  به  $t$  وجود دارد. الگوریتم پیشنهادی نیز مشابه همین الگوریتم طراحی شده است؛ با این تفاوت که برای مسأله‌ی گرگ و گوسفند تنها به دو تکرار احتیاج داریم تا دو مسیر به دست آیند. در زیر یک اجرا از برنامه آورده شده است:

```
C: > Users > Asus > Desktop > projects > wolf and st

15 g = [[0, 1, 1, 1, 0, 0, 0, 0],
16      [0, 0, 1, 0, 0, 0, 0, 0],
17      [0, 0, 0, 1, 0, 0, 1, 0],
18      [0, 0, 0, 0, 0, 0, 0, 1],
19      [0, 0, 1, 0, 0, 0, 0, 1],
20      [0, 1, 0, 0, 1, 0, 0, 1],
21      [0, 0, 0, 0, 0, 1, 0, 1],
22      [0, 0, 0, 0, 0, 0, 0, 0]]

TERMINAL PROBLEMS OUTPUT DEBUG CONSOLE
PS C:\Users\Asus> & C:/Users/Asus/anaconda3/
[0, 2, 6, 7]
[0, 3, 6, 5, 7]
```

برای زمان اجرای برنامه نیز دقت شود هر اجرای تابع BFS زمان  $O(n^2)$  می‌برد و چون حداکثر دو بار این تابع صدا زده می‌شود پیچیدگی زمانی کل برنامه نیز  $O(n^2)$  است.