

به نام خدا

- تحلیل مسئله کامیون و بسته:

الگوریتم به این صورت می باشد که همه کامیون ها را در یک صف اولویت بر اساس وزنشان نگهداری می نماییم و موقع اختصاص بسته ها به سبک ترین کامیون بسته اختصاص می دهیم. ( رویکرد حریصانه)

```
><4 go setup calls>
Enter the number of box:
6
Enter weight of box 1: 10
Enter weight of box 2: 20
Enter weight of box 3: 12
Enter weight of box 4: 30
Enter weight of box 5: 40
Enter weight of box 6: 50
Enter number of Truck
3
Truck id: 0, Truck weight: 40
Truck id: 1, Truck weight: 52
Truck id: 2, Truck weight: 70
Max weight on the Truck 2: 70

Process finished with exit code 0
|
```

- تحلیل مسئله گرگ و گوسفند:

یک از کاربردهای الگوریتم maxflow محاسبه نمودن بیشترین تعداد مسیرهایی است که دارای یال مشترک بین دو گره s و t در یک گراف جهت دار نمی باشند.

برای حل کردن این مسئله با استفاده از این الگوریتم کافی می باشد که برای ظرفیت هر یال در این گراف، مقدار 1 را در نظر بگیریم و الگوریتم را روی آن اجرا نماییم. اگر نتیجه ما بزرگتر مساوی 2 بود یعنی ما بیش از دو مسیر جدا داریم.

علت آن است که همیشه flow مسیر از 1 بیشتر نمی شود و باید این مقدار را از کل مسیر کم نماییم. این گونه از گراف ما یک مسیر حذف می شود و مسیرهای دیگر باقی می ماند و الگوریتم را روی آن دوباره اجرا می نماییم.

اگر دو مسیر وجود داشت، 1+1 می شود و برابر با 2 می شود که نمایانگر آن است که دو مسیر جدا وجود دارد.

```
Enter number of vertices and edges:
```

```
4 4
```

```
Enter (u v)
```

```
0 1
```

```
Enter (u v)
```

```
0 2
```

```
Enter (u v)
```

```
1 3
```

```
Enter (u v)
```

```
2 3
```

```
Enter source and sink
```

```
0 3
```

```
Sheep will have a nice trip!
```

```
Steps:
```

```
0
```

```
1
```

```
3
```

```
Steps:
```

```
0
```

```
2
```

```
3
```

بله، maxflow یک گراف با minCut آن برابر می باشد.

پیچیدگی زمانی از آنجا که ما در این پیاده سازی از ماتریس مجاورت استفاده کرده ایم و بر روی آن BFS زده ایم در اینجا برابر با  $V^2$  است. از طرفی تعداد iteration های الگوریتم ما، EV است. در نتیجه پیچیدگی زمانی برابر با  $EV^3$  است.