ریحانه حیدری 9731483

پروژه گرگ و گوسفند

شرح مسئله:

یک گرگ و یک گوسفند داریم که می خواهیم در یک گراف جهت دار و بدون دور از یک راس مثلا s به راس دیگری همچون t بروند، برای اینکه از خورده شدن گوسفند توسط گرگ جلوگیری کنیم، مسیر این دو نباید هیچ یال مشترکی داشته باشد.

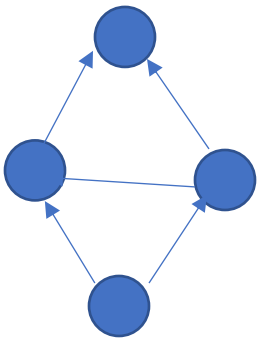
در این پروژه می خواهیم یک الگوریتم با پیچیدگی چند جمله ای ارائه دهیم که دو مسیر با این ویژگی (جدا از یال - edge-disjoint) را پیدا کند، در صورتی که چنین مسیری وجود داشت (امنیت گوسفند تضمین شود)، این مسیر ها را چاپ کند.

روش حل:

ابتدا با استفاده از dfs یک مسیر از s به t پیدا می کنیم و سپس تمامی خانه های آرایه ی mark را صفر می کنیم.

حال اگر بخواهیم دوباره از sبه t dfsبزنیم اما از یالهای مسیر قبل استفاده نکنیم (آنها را از گراف حذف کرده باشیم) ممکن است دو مسیر مجزای یالی در گراف وجود داشته باشد اما مسیر اولی که در dfs پیدا شد از یال های هر مسیر در خود داشته باشدکه در نتیجه با حذف آنها دیگر مسیر از s به t باقی نمانده باشد.

مثال:



اگر راس پایین را s و راس بالا را t و راس راست و چپ را 1 و 2 درنظر بگیریم.

مسیر اول:

(s,1)(1,2)(2,t)

دو مسیر مجزا یالی:

(s,1)(1,t)

(s,2)(2,t)

برای حل این مشکل بعد از پیدا کردن

رابطه مسئله با شار بیشینه:

این مسئله وقتی یک مسیر جدید از s به t پیدا میکنیم مثل این می ماند که می توانیم یک جریان را از s به t بگذرانیم.

در این مسئله دو مسیر مجزا با این شرایط می یابیم که معادل دو شار یا جریان در مسئله ی max flow هست.(برای مثال گنجایش هر یال را میتوانیم 1 درنظر بگیریم) و میدانیم که خود مسئله ی max flow نیز معادل min cut می باشد.

تحلیل پیچیدگی زمانی:

برابر با دوبار اجرای dfs است که از O(n+m) است.

می شود که در بعضی جاها لیست mark را برابر صفر کنیم که آن هم می تواند تا n انجام شود.

مرتبه زمانی الگوریتم=O(n+m)

عملکرد برنامه: