

به نام خدا

گزارش پروژه چهارم (گرگ و گوسفند) غزل پوراسفندیار بروجنی (9820453)

بناست در گراف جهت دار مسیر هایی بین دو راس مشخص بیابیم به طوری که این مسیرها یال مشترک نداشته باشند. ابتدا تعداد مسیرهایی که می توان یافت را مشخص می کنیم. این مساله را میتوان به مساله شار بیشینه کاهش داد. به طوری که ظرفیت هر یال برابر 1 باشد و این به این معناست که هر یال فقط در یک مسیر می تواند باشد. پس شار بیشینه به دست آمده، ماکزیمم تعداد مسیرهای ممکن است. (ممکن است مسیرها یکتا نباشند. ولی تعدادشان یکتاست.) مساله یافتن شار بیشینه را با الگوریتم فوردفولکرسون حل میکنیم.

اگر تعداد مسیرهای بدون یال مشترک (شار بیشینه) بزرگتر یا مساوی 2 باشد، امنیت گوسفندان تضمینی است و آنگاه حداقل دو مسیر پیشنهادی ارائه می کنیم.

برای ارائه مسیر های پیشنهادی کافیهست یک مسیر از مبدا به مقصد را بیابیم، و سپس یالهای مسیر پیدا شده را از گراف باقی مانده حذف کنیم و مجددا الگوریتم یافتن مسیر را به تعداد ماکزیمم مسیرهای ممکن تکرار کنیم. در این کد، از الگوریتم BFS برای یافتن کوتاهترین مسیر در از مبدا تا مقصد استفاده شده است.

پیچیدگی زمانی الگوریتم:

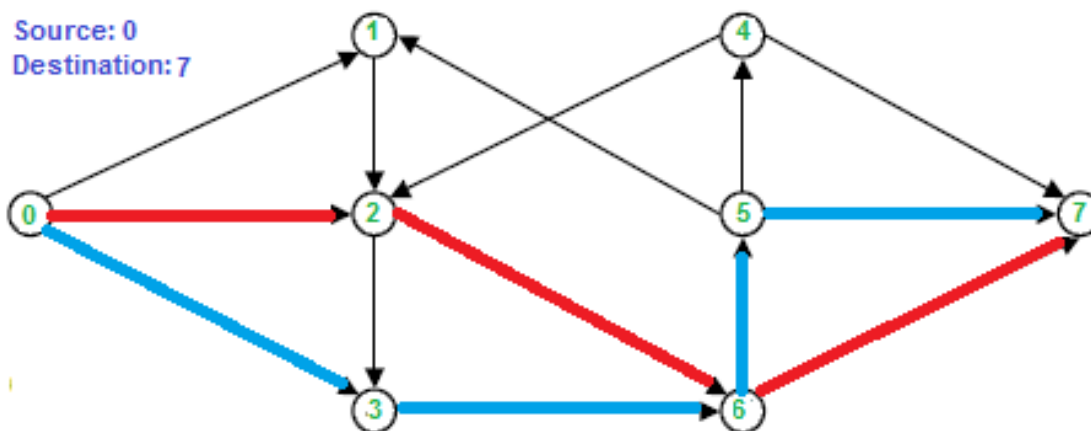
همه الگوریتم ها و توابع نوشته شده در مرتبه زمانی چند جمله ای کار میکنند و لذا پیچیدگی زمانی برنامه، از مرتبه چند جمله ای می باشد.

پیچیدگی زمانی الگوریتم فورد فولکرسون : $O(nm^2)$

پیچیدگی زمانی الگوریتم پیدا کردن مسیر از s به t : $O(n + m)$

مثالی از ورودی برنامه :

گراف زیر در تابع initialAdj() در کلاس Main به وسیله لیست مجاورت (آرایه ای از لیست های حاوی یالها) مقداردهی شده است. و بناست از راس 0 به راس 7 مسیرهای بدون یال مشترک را در صورت وجود، بیابیم.



مثالی از خروجی برنامه :

خروجی برنامه لزوماً یکتا نیست. یک نمونه خروجی در شکل بالا با دو رنگ نمایش داده شده است. اما مثلاً میتوانستیم به جای مسیر آبی رنگ، مسیر 7 -> 5 -> 6 -> 3 -> 2 -> 1 -> 0 را هم انتخاب کنیم.

خروجی برنامه برای این گراف به صورت زیر خواهد بود :

```
Maximum number of disjoint edge paths is : 2.0  
The path number 1 is : 0 2 6 7  
The path number 2 is : 0 3 6 5 7
```

این به این معناست که ماکزیمم 2 مسیر بدون یال تکراری خواهیم داشت و دو مسیر موجود هم ذکر شده است.