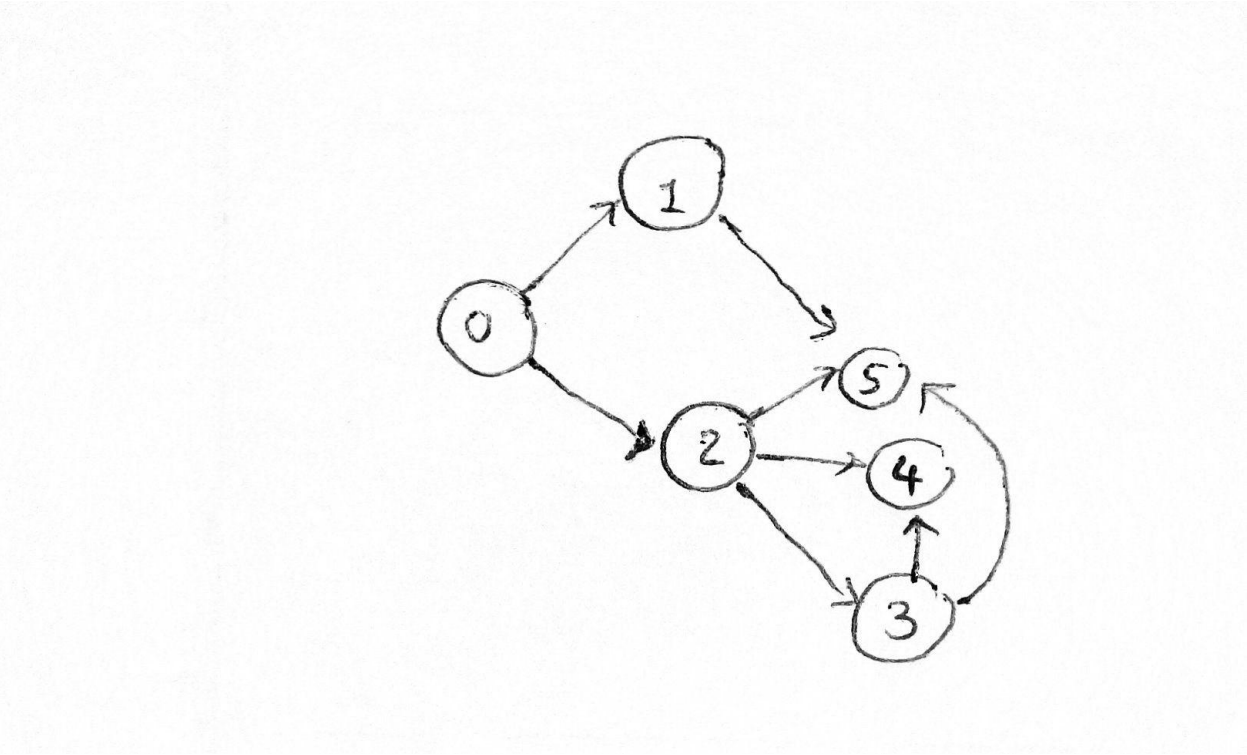
بنام خدا

پروژه ی چهارم درس طراحی الگوریتم (گرگ و گوسفند)

در این مسئله می خواهیم دو مسیر با یال های جدا از هم پیدا کنیم به گونه ای که امنیت گوسفند تضمین شود.
این مسئله مشابه مسئله ی پیدا کردن min-cut و max-flow در یک گراف جهت دار می باشد و با کاهش این مسئله به این مسائل قابل ما می توانیم ماکزیمم تعداد مسیر های جدا از هم را با استفاده از الگوریتم های موجود برای پیدا کردن max-flow بیابیم (همانطور که در کلاس نشان داده شد min-cut هم معادل max-flow می باشد.)
در واقع اگر از روش ادموندز-کارپ که برای بهبود الگوریتم فورد فولکرسون پیشنهاد کرده بود که برای پیدا کردن مسیرهای افزاینده بجای استفاده از dfs یا یک مسیر کاملا رندم از bfs استفاده شود، استفاده کنیم، مسیر های افزاینده ای که در هر مرحله پیدا می شود، یال مشترکی ندارند و شرایط خواسته شده در مسئله ی ما را دارند، در واقع در اینجا مقدار ظرفیت هر یال یک می باشد و با هر بار پیدا کردن مسیر افزایشی از گراف باقیمانده ی ما حذف می شود.
در این مسئله برای پیدا کردن دو مسیر جدا از هم روند کار ما به این صورت می باشد که مشابه روش ادموندز-کارپ از bfs استفاده می کنیم و کوتاه ترین مسیر بین ابتدا و انتهای مسیر را می یابیم، سپس مسیر پیدا شده را از گراف مجاورت حذف می کنیم و دوباره از bfs برای پیدا کردن کوتاه ترین مسیر بین مبدأ و مقصد استفاده می کنیم، در صورتی که bfs مسیری بین مبدأ و مقصد ما پیدا کند مسئله ی ما حل می شود در غیر این صورت می گوییم چنین مسیری وجود ندارد.

پیچیدگی زمانی
برای حل این مسئله ما فقط دوبار از bfs استفاده کردیم و برای پیاده سازی آن از صف استفاده کردیم درنتیجه پیچیدگی زمانی ما همان پیچیدگی زمانی bfs یعنی O(V+E) می باشد.

نمونه ای از اجرای برنامه :




```
6 8
0 1
0 2
1 5
2 3
2 4
2 5
3 4
3 5
path:0->1->5->[0, 1, 5]

path:0->2->5->[0, 2, 5]


Process finished with exit code 0
```