

بسم الله الرحمن الرحيم

9729293

علی قیومی

گزارش پروژه چهارم طراحی الگوریتم-گرگ و گوسفند

توضیح کلی و ارتباط با شار بیشینه:

برای پیدا کردن مسیرهای بدون یال مشترک می‌توانیم از الگوریتم‌های یافتن شار بیشینه به عنوان مثال فورد-فالکرسون استفاده کنیم؛ بدین شکل که ظرفیت هر یال را 1 در نظر می‌گیریم. در هر بار اجرای این الگوریتم ظرفیت یالی که در مسیر هست به اندازه‌ی مینیمم شار مسیر کم می‌شود. در این حالت چون ظرفیت‌ها یک هست ظرفیت یال منفی شده و دیگر در گام‌های بعد آن یال طی نخواهد شد.

توضیح کد:

کلاس گراف برای پیاده‌سازی گراف با ماتریکس مجاورت نوشته شده است. در لیست disjointPaths مسیرها نگه داری خواهند شد.

متدهای addEdge و hasEdge و setAdjacentMatrix واضح هستند.

متودهای setWeight و getWight برای اضافه کردن وزن به گراف باقی‌مانده‌ی الگوریتم فورد-فالکرسون نوشته شده اند.

متود bfs در گراف با الگوریتم bfs پیمایش می‌کند اگر به راس مقصد رسیدیم مقدار صحیح و در غیر این صورت مقدار ناصحیح برمی‌گرداند. مسیری که طی می‌شود در آرایه‌ی path ذخیره خواهد شد. به این صورت که $path[v]=u$ به این معنی است که از راس u به v رفته‌ایم.

متود findDisjointPath الگوریتم فورد-فالکرسون را اجرا می‌کند با این تفاوت که بعد از هر گام متود addToPaths را صدا می‌زنیم تا بررسی شود این مسیر در گراف اصلی هم وجود دارد یا خیر. اگر وجود داشت مسیر را از آرایه استخراج کرده و به disjointPaths اضافه می‌کنیم. در پایان الگوریتم اگر شار بیشینه بیش از یک بود به معنی این است که مسیر با ویژگی گفته شده وجود دارد پس مسیرها را چاپ می‌کنیم.

پیچیدگی زمانی:

پیچیدگی زمانی الگوریتم همان پیچیدگی زمانی الگوریتم فورد-فالکرسون یعنی $O(\maxFlow * E)$ می‌باشد. در این جا از پیاده‌سازی‌ای از این الگوریتم به اسم ادموند-کارپ استفاده شده است که از bfs استفاده می‌کند و پیچیدگی زمانی به $O(VE^2)$ کاهش می‌یابد. هر بار اجرای bfs $O(E)$ طول می‌کشد چون که هر راس حداقل یک یال دارد. حلقه‌ی while الگوریتم نیز حداکثر VE بار اجرا خواهد شد.

```
Enter number of vertexes:
6
enter new Edge or -1 for finish
0
1
enter new Edge or -1 for finish
1
2
enter new Edge or -1 for finish
2
5
enter new Edge or -1 for finish
0
3
enter new Edge or -1 for finish
3
4
enter new Edge or -1 for finish
4
5
enter new Edge or -1 for finish
0
5
enter new Edge or -1 for finish
-1
Enter beginnig:0
Enter destination:5
0 5
0 1 2 5
0 3 4 5
```