



گزارش پروژه نهایی

درس طراحی الگوریتم

مسئله «گرگ و گوسفند»

مجتبی فیاضی – 9728973

توضیحات کلی

با داشتن یک گراف و دو رأس s و t ، هدف پیدا کردن دومسیر (در صورت امکان) از رأس s به رأس t با یال‌های مجزاست. این مسئله یک حالت خاص از مسئله شار بیشینه (maximum flow-minimum cut) است که در آن ظرفیت یال‌ها 1 است.

با داشتن گراف جهت‌دار $G=(V,E)$ و رئوس مشخص s و t ، به صورت زیر یک شار در این گراف تعریف می‌کنیم. به هر یک از یال‌های این گراف ظرفیت 1 اختصاص می‌دهیم. می‌توان اثبات کرد که تعداد مسیرهای edge-disjoint از s به t در این گراف، برابر شار بیشینه گراف است. بنابراین برای حل مسئله کافی است شار بیشینه گراف داده شده را بدست بیاوریم، اگر بزرگتر یا مساوی 2 بود، آنگاه مسئله جواب دارد و در غیر این صورت می‌گوییم مسئله جواب ندارد.

برای حل این مسئله و پیدا کردن شار بیشینه، از الگوریتم Edmonds-Karp استفاده شده است. این الگوریتم مشابه ولی بهینه‌تر از الگوریتم فورد-فالکرسون است؛ زیرا در آن برای پیدا کردن مسیرهای augmented، از جست‌وجوی BFS استفاده می‌شود. برای اینکه مسیرهای یافته شده را چاپ کنیم، در حین کار این الگوریتم مسیرها نیز ذخیره می‌شوند و در نهایت در صورت وجود دو مسیر مطلوب، در خروجی نمایش داده می‌شوند.

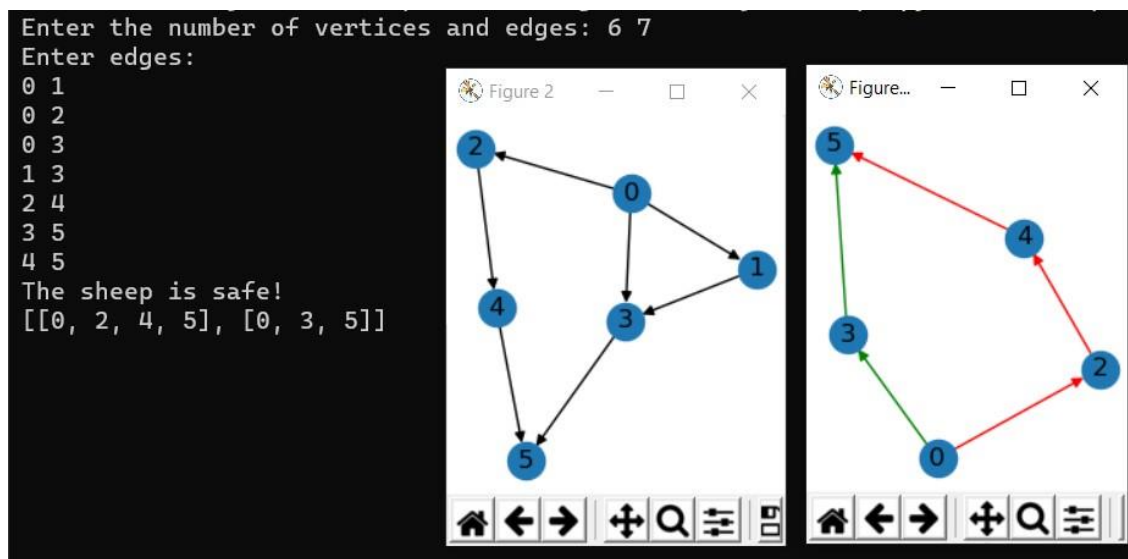
```
def getDisjointPaths(self, source, sink):
    parent = [-1]*(self.nodes)
    max_flow = 0
    paths = ['#']
    # Augment the flow while there is path from source to sink
    while self.BFS(source, sink, parent):
        path_flow = float("Inf")
        s = sink
        paths.append(s)
        while(s != source):
            path_flow = min(path_flow, self.graph[parent[s]][s])
            s = parent[s]
            paths.append(s)

        paths.append('#')
        max_flow += path_flow

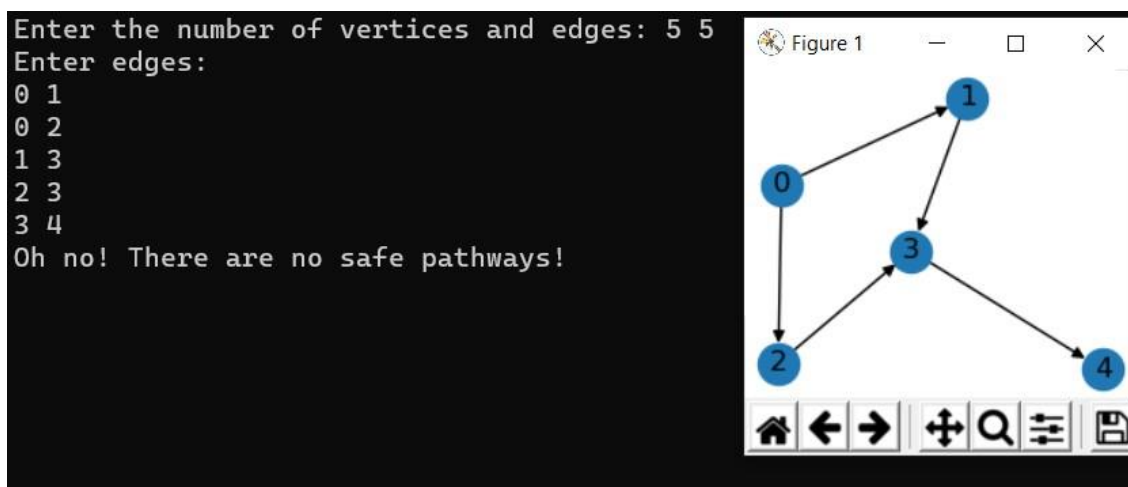
        v = sink
        while(v != source):
            u = parent[v]
            self.graph[u][v] -= path_flow
            self.graph[v][u] += path_flow
            v = parent[v]

    return paths, max_flow
```

شکل زیر یک نمونه اجرای موفق الگوریتم نشان داده شده است. در این کد، همواره رأس صفرم همان source و رأس 1-V ام همان sink در نظر گرفته شده است.



یک نمونه اجرای ناموفق:



پیچیدگی زمانی

پیچیدگی زمانی این الگوریتم برابر پیچیدگی زمانی الگوریتم Edmonds-Karp است که یک الگوریتم با پیچیدگی زمانی چندجمله‌ای است. در گرافی با V رأس و E یال، پیچیدگی زمانی الگوریتم بالا برابر است با:

$$O(V*(E^2))$$