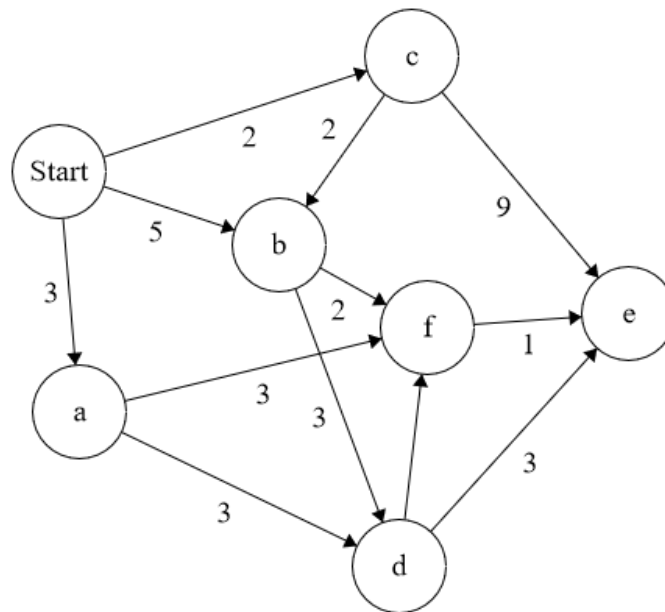
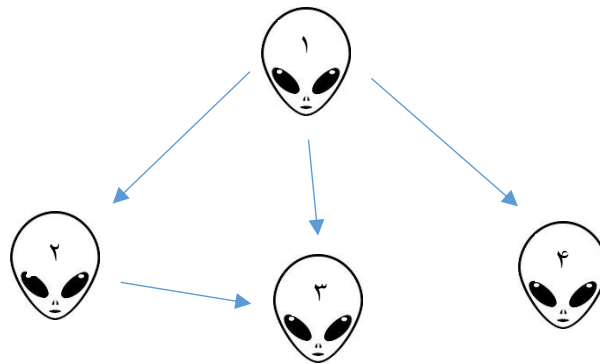


۱) گراف جهت دار زیر را در نظر بگیرید. برای این گراف ابتدا پیمایش BFS و DFS را رسم کنید و سپس با در نظر گرفتن وزن یال ها پیمایش Dijkstra را نیز برای آن رسم کنید. در نظر داشته باشید که در شرایط برابر برای پیمایش به ترتیب حروف الفبا پیمایش انجام دهید.



۲) در طی بررسی های جدید دانشمندان به سیاره ای جدید در منظومه شمسی رسیده اند که موجودات جدیدی به نام «تکراری» در آن زندگی می کنند. این موجودات برای خودشان یک شبکه اجتماعی خاص دارند که به آن «بتکرار» می گویند. در این شبکه اجتماعی هر فرد مجموعه ای افراد را دنبال می کند و توسط مجموعه ای دنبال می شود. اگر فرد A فرد B را در این شبکه اجتماعی دنبال کند به این معنی است که هر محتوایی که توسط B گفته شود توسط A بازنشر می شود. و این روند تا پایان ادامه دارد. در گراف شکل زیر در صورتی که A فرد B را دنبال کند یالی از B به A کشیده شده است.



در این سیاره به فردی که در صورت گفتن چیزی به همه منتقل شود لقب «مکرر» را می دهند. برای مثال در شکل بالا فرد ۱ مکرر است. (در نظر بگیرید تعداد اعضای این سیاره n و تعداد یال ها m است.)

الف) ثابت کنید اگر تمام افراد مکرر در این سیاره را در نظر بگیریم، بین هر دوی آنها یک مسیر وجود دارد.

ب) فرض کنید حداقل یک مکرر در این جزیره وجود دارد. حال الگوریتمی از مرتبه $O(n+m)$ ارائه دهید که بتوانید یکی از مکررها را پیدا کنید.

ج) فرض کنید نمی دانیم که آیا در این جزیره مکرر وجود دارد یا خیر. الگوریتمی ارائه کنید که در مرتبه زمانی $O(n+m)$ در صورت وجود یک مکرر را معرفی کند و در غیر این صورت اعلام کند که مکرری وجود ندارد.

۳) الگوریتم **dijkstra** زیر را در نظر بگیرید :

```
def dijkstraShortestPath(G,s,t):
    d[v] = Infinity for all v in G.vertices
    d[s] = 0
    unsureVertices = G.vertices
    while len(unsureVertices) > 0:
        u = a vertex in unsureVertices so that d[u] is minimized
        if d[u] == Infinity:
            break
        for v in u.getOutNeighbors(): //all v are in unsureVertices
            if d[u] + weight(u,v) < d[v]:
                d[v] = d[u] + weight(u,v)
                v.parent = u
        unsureVertices.remove(u)

    if d[t] == Infinity:
        return "Can't reach t from s!"
    path = []
    current = t
    while current != s:
        path.append(current)
        current = current.parent
    path.append(current)
    path.reverse()
    return path
```

الف) در صورتی که وزن یال ها بزرگتر مساوی 0 باشند. تغییر دادن شرط $d[u] + \text{weight}(u,v) < d[v]$ به $d[u] + \text{weight}(u,v) \leq d[v]$ می تواند باعث بروز چه مشکلی شود؟

(مثالی بزنید و توضیح دهید چرا این الگوریتم مسیر درستی نمی دهد)

ب) در صورتی که وزن یال ها بتواند هر عدد صحیح باشد مثالی از یک گراف بزنید که در آن مسیر از s به t وجود داشته باشد اما کوچکترین مسیر وجود نداشته باشد.

(مثالی بزنید و بگویید چرا کوچکترین مسیر وجود ندارد)

ج) در صورتی که وزن یال ها بتواند هر عدد صحیح باشد. مثالی بزنید که کوچکترین مسیر وجود داشته باشد اما الگوریتم بالا نتواند آن را پیدا کند.

(مثالی بزنید و توضیح دهید چرا الگوریتم بالا کوچکترین مسیر را نمی یابد)

د) به ازای هر گراف $G=(V,E)$ اندازه کوچکترین یال این گراف را w' می نامیم. حال گراف جدید $G'=(V,E')$ را به این صورت می سازیم که به ازای هر یال درون E با اندازه w یالی با اندازه $w-w'$ به این گراف اضافه می کنیم. حال ادعا می کنیم با استفاده از این ورودی جدید می توانیم مشکل یال های منفی را حل کنیم. آیا ادعای ما صحیح است؟ مثال بزنید.