

## سید علی تولیت - ۹۸۲۰۷۱۳ - پروژه طراحی الگوریتم ۴ - استاد احمدی

رویکرد برنامه:

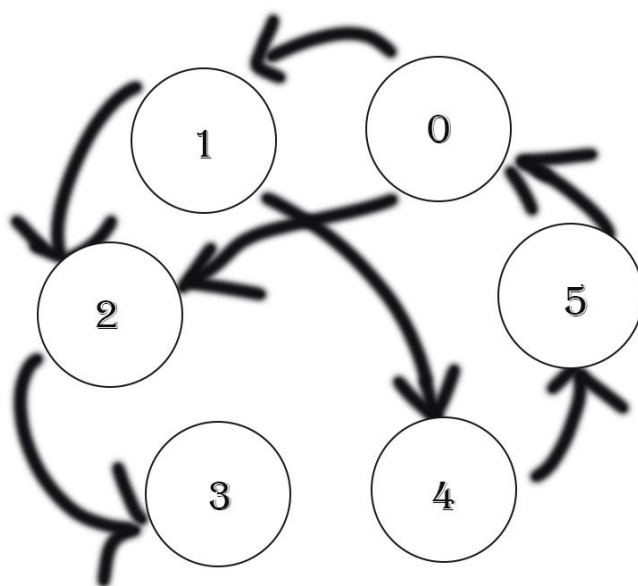
ایده این است که DFS را روی گراف اجرا می کنیم. (از مبدا شروع می کنیم)

راسهای بازدید شده را در یک آرایه ذخیره می کنیم و در صورتی که به مقصد رسیدیم مسیر طی شده را از راسهای داخل آرایه به دست می آوریم، در اینجا بررسی میکنیم که آیا در اعضای مسیر یکی از رئوس همسایه با راس ممنوعه (گرگ) پیدا می کنیم یا نه که اگر نباشد؛ آنگاه مسیر، مسیر مطلوب است.

(از آن جایی که هم که گراف حلقه ندارد پس نیاز به علامت گذاری رئوس نیست.)

پیچیدگی زمانی نیز چند جمله ای است و از هر راس  $V$  رئوس وجود دارد که می توان از راس فعلی بازدید کرد:  $O(V^V)$

مثال:



پس از مقدار دهی گراف:

```
number of vertices:  
6  
vertices from 0 to 5 initialized.
```

```
add edges:(to stop enter -1)
```

```
from:
```

```
0
```

```
to:
```

```
1
```

```
from:
```

```
1
```

```
to:
```

```
2
```

```
from:
```

```
2
```

```
to:
```

```
3
```

```
from:
```

```
0
```

```
to:
```

```
2
```

```
from:
```

```
1
```

```
to:
```

```
4
```

```
from:
```

```
4
```

```
to:
```

```
5
```

```
from:
```

```
5
```

```
to:
```

```
0
```

```
from:
```

```
-1
```

```
to:
```

```
-1
```

```
find path from:
```

```
0
```

```
to:
```

```
3
```

```
forbidden vertice:
```

```
4
```

پاسخ به صورت زیر است:

Following are all validated paths from 0 to 3

[۳,۲,۰]