

## به نام خدا

کلاس graph حاوی 2 تابع است:

- تابع `__init__` که شامل مقدار دهی اولیه به گراف است.
- تابع `print_of_pathes` تمام مسیرها بین دو راس شروع و پایان را محاسبه و چاپ می کند.
- لیست `visited` راس های تاکنون دنبال شده را ذخیره می کند و لیست `pathes` جهت ذخیره مسیر طی شده است.
- پیچیدگی زمانی از مرتبه  $O(V^V)$  خواهد بود. ( $V$  یال است)
- برای الگوریتم عقب گرد دو شرط وجود یال بین دو راس و یکبار عبور کردن از راس باید رعایت شود.
- الگوریتم: ابتدا راس ورودی به `visited` اضافه می شود و به `Pathes` اضافه می شود. سپس اگر برابر راس مقصد بود مسیر چاپ می شود در غیر این صورت همین روال برای راس هایی که مجاور راس ورودی هستند تکرار میشود. در انتها راس فعلی هم از `visited` و `path` حذف می شود.
- عکس اجرا برنامه:

```
6
1 2
1 3
3 4
3 5
4 6
5 6
2 3
2 4
0 0
4
2 3
3 4
5 1
1 6
7 8
8 9
2 5
5 7
3 1
1 8
4 6
6 9
0 0
```



```
CASE 1:
[1, 2, 3, 4, 6]
[1, 2, 3, 5, 6]
[1, 2, 4, 3, 5, 6]
[1, 2, 4, 6]
[1, 3, 4, 6]
[1, 3, 5, 6]
[1, 3, 2, 4, 6]
CASE 2:
[1, 5, 2, 3, 4]
[1, 5, 7, 8, 9, 6, 4]
[1, 6, 4]
[1, 6, 9, 8, 7, 5, 2, 3, 4]
[1, 3, 2, 5, 7, 8, 9, 6, 4]
[1, 3, 4]
[1, 8, 7, 5, 2, 3, 4]
[1, 8, 9, 6, 4]
```