به نام خدا

تابع init نیز جهت مقداردهی اولیه به object گراف استفاده می شود.

تابع addEdge یال ها را براساس رای شروع و انتها به گراف ساخته شده اضافه میکند.

تابع printAllPathsUtil 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 path جهت ذخیره مسیر طی شده است.

الگوریتم : ابتدا راس ورودی به visited اضافه می شود و به Path اضافه می شود. سپس اگر برابر راس مقصد بود مسیر چاپ می شود درغیر 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.

*** برای الگوریتم عقب گرد دو شرط لازم است:

وجود یال بین دو راس و یکبار عبور کردن از هر راس

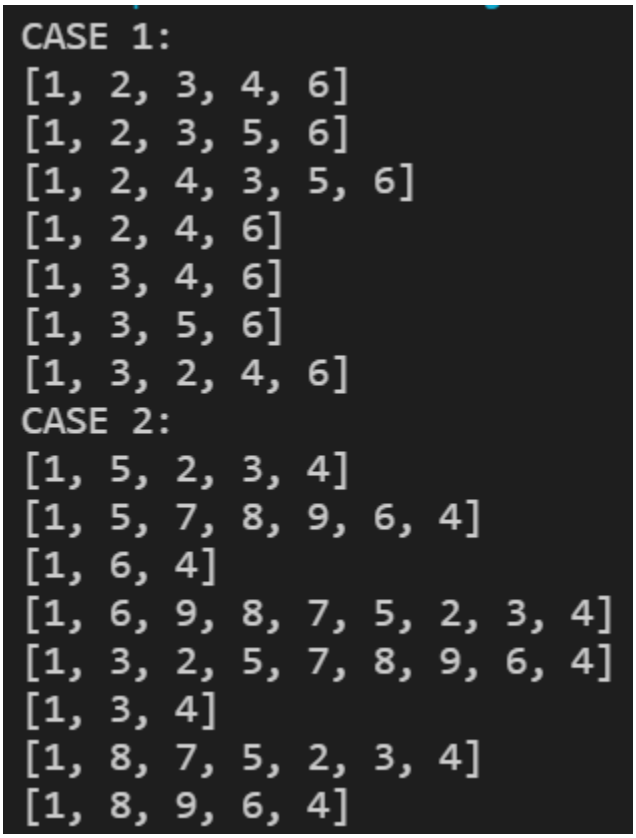
پیچیدگی زمانی از مرتبه O(V^V) خواهد بود.

نمونه ورودی و خروجی:

INPUT:

```
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
```

OUTPUT:


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]