

# Permutation Game (permgame)

Alice 同 Bob 係細個識到大嘅朋友，佢哋好鍾意玩燒腦嘅 game。今日佢哋就喺 graph 上面玩一隻新 game。

隻 game 有一個 **connected** 嘅 graph，有  $m$  個由 0 去到  $m - 1$  嘅 vertex，同埋  $e$  條由 0 去到  $e - 1$  嘅 edge。第  $i$  條 edge 連住  $u[i]$  同  $v[i]$  呢兩個 vertex。

隻 game 仲有一個 permutation  $p[0], p[1], \dots, p[n - 1]$ ，長度係  $n$ ，而且  $m \leq n$ 。Permutation 係一個 array，入面每一個由 0 去到  $n - 1$  嘅數都啱啱好出現一次，順序就有所謂。個 permutation  $p$  嘅 **分數** 就係有幾多個 index  $i$  符合  $p[i] = i$ 。

隻 game 最多會行  $10^{100}$  步。每一步，就會發生以下嘅嘢：

1. 如果 Alice 決定終止隻 game，隻 game 就會停。
2. 如果唔係，Alice 就會揀 **一啲唔同嘅 index**  $t[0], t[1], \dots, t[m - 1]$ ， $0 \leq t[i] < n$ 。留意番，隻 game **唔要求**  $t[0] < t[1] < \dots < t[m - 1]$ 。
3. Bob 就會喺個 graph 度揀  $0 \leq j < e$  號呢條 edge，跟住將  $p[t[u[j]]]$  同  $p[t[v[j]]]$  對調。

Alice 想個 permutation 最後嘅分數最高，Bob 就想佢最低。

你個任務就係幫 Alice 打低 Bob，Bob 嘅動作由會個 grader 模擬。

我哋先定義 **最佳分數** 做如果 Alice 同 Bob 都用最佳策略玩遊戲，最後個排列嘅分數會係幾多。

你需要先判斷個排列嘅最佳分數，然後再同 Bob 玩呢個遊戲，喺玩咗幾輪之後，**至少** 擺到等於個最佳分數嘅分數。

留意番，Alice 嘅策略無論 Bob 點行都應該 work，就算 Bob 唔用最佳策略都係噉話。

## 點樣 Implement

你要寫以下呢個 procedure：

```
int Alice(int m, int e, std::vector<int> u, std::vector<int> v,
          int n, std::vector<int> p)
```

- $m$ : 個 graph 有幾多個 vertex。
- $e$ : 個 graph 有幾多條 edge。
- $u$  同  $v$ : 兩個 array，長度都係  $e$ ，用黎表示個 graph 嘅 edge。

- $n$ : 個 permutation 嘅長度。
- $p$ : 一個 array，長度係  $n$ ，就係個 permutation。
- 呢個 procedure 淨係會 call 啱啱好一次。
- 呢個 procedure 要 return 一個整數 – 隻 game 嘅最佳分數。

喺呢個 procedure 入面，你可以 call 以下呢個 procedure：

```
int Bob(std::vector<int> t)
```

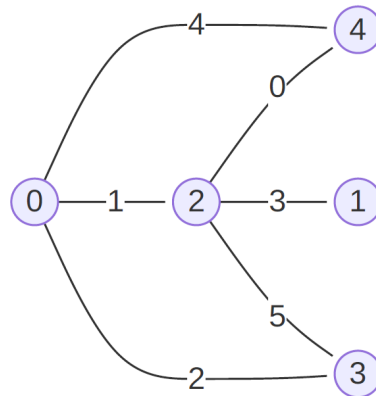
- $t$ : 一個 array，size 係  $m$ ，入面係啲唔同嘅 index， $0 \leq t[i] < n$  而且對於所有  $i \neq j$ ， $t[i] \neq t[j]$ 。
- 呢個 function 會 return 一個整數  $j$ ， $0 \leq j < e$ 。
- 呢個 procedure 可以 call 好多次。

## 例子

如果有以下呢個 call：

```
Alice(5, 6, [4, 0, 3, 1, 4, 2], [2, 2, 0, 2, 0, 3],  
      10, [8, 2, 7, 6, 1, 5, 0, 9, 3, 4])
```

個 graph 係咁樣嘅：



而  $p$  一開始就係  $[8, 2, 7, 6, 1, 5, 0, 9, 3, 4]$ 。

根據上面嘅限制，我哋可以證明，呢個 permutation 嘅最佳分數係 1。

假設，Alice 行以下呢 4 步：

昇 Bob 嘅 $t$	Bob return 個數	對應 $p$ 嘅 index	Bob swap 完之後嘅 $p$
[3, 1, 5, 2, 0]	5	5, 2	[8, 2, 5, 6, 1, 7, 0, 9, 3, 4]
[9, 3, 7, 2, 1]	0	1, 7	[8, 9, 5, 6, 1, 7, 0, 2, 3, 4]
[5, 6, 7, 8, 9]	1	5, 7	[8, 9, 5, 6, 1, 2, 0, 7, 3, 4]
[7, 5, 2, 3, 6]	3	5, 2	[8, 9, 2, 6, 1, 5, 0, 7, 3, 4]

留意番上面 Alice 同 Bob 未必會行出最佳嘅步數。呢啲步數純粹為咗示範。仲有，留意番 Alice 可以即刻完咗個遊戲，因為個 permutation 嘅初始分數已經係 1。

Alice 行晒上面啲步之後，個 permutation 嘅實際分數係 3 ( $p[2] = 2, p[5] = 5, p[7] = 7$ )。

最後，個 `Alice()` function 會 return 1，就係 permutation 嘅最佳分數。

留意番，就算 Alice 同 Bob 玩到有 3 分，如果個 `Alice()` return 番 3 而唔係 1，你都會攤 0 分。

## 限制

- $2 \leq m \leq 400$
- $m - 1 \leq e \leq 400$
- $0 \leq u[i], v[i] < m$
- $m \leq n \leq 400$
- $0 \leq p[i] < n$
- 個 graph 係 connected 嘅，冇 self-loop 亦都有 multiple edge。
- $p$  係一個 permutation，即係話如果  $i$  同  $j$  唔同，咁  $p[i]$  同  $p[j]$  都唔同。

## Subtask

1. (6 分)  $m = 2$
2. (6 分)  $e > m$
3. (10 分)  $e = m - 1$
4. (24 分)  $e = m = 3$
5. (24 分)  $e = m = 4$
6. (30 分)  $e = m$

每一個 subtask，你都可以攤部分分數。假設  $r$  係啲呢個 subtask 所有 test case 入面，最高嘅  $\frac{k}{n}$  比例， $k$  就係行咗幾多步（即係 `call Bob()` 嘅次數）。咁樣，你啲嗰個 subtask 嘅分數就會乘以以下呢個數：

條件	乘以
$12 \leq r$	0
$3 < r < 12$	$1 - \log_{10}(r - 2)$
$r \leq 3$	1

特別係，如果你喺  $3n$  步之內解到題，你嗰個 subtask 就會攞滿分。用多過  $12n$  步嘅話，你喺嗰個 subtask 就會攞 0 分（會顯示 `Output isn't correct`）。

## Sample Grader

個 sample grader 會讀以下呢個格式嘅 input：

- 第一行： $m\ e$
- 第  $2 + i$  行 ( $0 \leq i \leq e - 1$ )： $u[i]\ v[i]$
- 第  $2 + e$  行： $n$
- 第  $3 + e$  行： $p[0]\ p[1]\ \dots\ p[n - 1]$

個 sample grader 根據以下格式 output：

- 第一行：最後個 permutation  $p$
- 第二行：`Alice()` return 番個數
- 第三行：最後個 permutation 嘅實際分數
- 第四行：行咗幾多步