排列遊戲

愛麗絲和鮑伯是從小一起長大的朋友，他們熱愛玩益智遊戲。今天，他們要在圖形上玩一個新遊戲。

遊戲配置包含一個**連通**圖，共有 m 個頂點（編號從 0 到 $m-1$）和 e 條邊（編號從 0 到 $e-1$）。第 i 條邊連接頂點 $u[i]$ 和 $v[i]$。

遊戲配置還包含一個長度為 n 的排列 $p[0], p[1], \ldots, p[n-1]$，其中 $m \leq n$。排列是指一個陣列，其中每個數字從 0 到 $n-1$ 恰好出現一次，順序不限。排列 p 的**分數**是指滿足 $p[i] = i$ 的索引 i 的數量。

遊戲最多進行 10^{100} 回合。每回合的流程如下：

1. 如果愛麗絲決定結束遊戲，遊戲停止。
2. 否則，愛麗絲選擇**不同的索引** $t[0], t[1], \ldots, t[m-1]$，其中 $0 \leq t[i] < n$。注意，遊戲**不**要求 $t[0] < t[1] < \ldots < t[m-1]$。
3. 鮑伯選擇圖的一條邊的索引 $0 \leq j < e$，並交換 $p[t[u[j]]]$ 和 $p[t[v[j]]]$ 的值。

愛麗絲希望最大化排列的最終分數，而鮑伯則希望最小化排列的最終分數。

你的任務是幫助愛麗絲對抗由評分程式模擬的鮑伯。

我們定義*最佳分數*為當愛麗絲和鮑伯都採取最佳策略時排列的最終分數。

你需要確定排列的最佳分數，然後在與鮑伯的遊戲中，經過若干回合後達到**至少**該最佳分數。

注意，無論鮑伯採取什麼策略（包括非最佳策略），愛麗絲的策略都應有效。

實作細節

你應實作以下函式：

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

- m：圖中的頂點數量。
- e：圖中的邊數量。
- u 和 v：長度為 e 的陣列，描述圖的邊。
- n：排列的長度。
- p：長度為 n 的陣列，描述排列。
- 此函式僅被呼叫一次。

- 此函式應回傳一個整數——遊戲的最佳分數。

在此函式中，你可以呼叫以下函式：

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

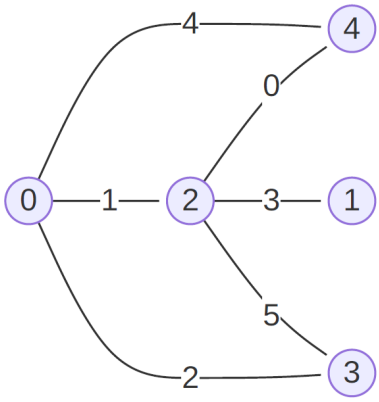
- t：大小為 m 的陣列，包含不同的索引，其中 $0 \le t[i] < n$ 且對於任何 $i \ne j$，$t[i] \ne t[j]$。
- 此函式回傳一個整數 j，滿足 $0 \le j < e$。
- 此函式可被多次呼叫。

範例

考慮以下呼叫：

```
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])
```

圖形如下：



且 p 初始為 $[8, 2, 7, 6, 1, 5, 0, 9, 3, 4]$。

根據上述限制，我們可以證明排列的最佳分數為 1。

假設愛麗絲進行以下 4 步操作：

傳遞給 Bob 的 t 參數	Bob 的回傳值	p 中對應的索引	鮑伯交換後的 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	$5, 2$	$[8, 9, 2, 6, 1, 5, 0, 7, 3, 4]$

注意，愛麗絲和鮑伯不一定採取最佳策略。這些操作僅供示範。另外，愛麗絲可以立即結束遊戲，因為排列的初始分數已經是 1。

在上述操作後，排列的實際分數為 3（$p[2] = 2, p[5] = 5, p[7] = 7$）。

最終，函式 `Alice()` 將回傳 1——排列的最佳分數。

注意，即使愛麗絲與鮑伯對戰後達到 3 分，如果 `Alice()` 回傳 3 而非 1，你將獲得 0 分。

限制條件

- $2 \le m \le 400$
- $m - 1 \le e \le 400$
- $0 \le u[i], v[i] < m$
- $m \le n \le 400$
- $0 \le p[i] < n$
- 圖是連通的，不包含自環或重複邊。
- p 是一個排列，即對於任何 $i \ne j$，$p[i] \ne p[j]$。

子任務

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) $e = m$

對於每個子任務，你可以獲得部分分數。設 r 為該子任務所有測試案例中 $\frac{k}{n}$ 的最大比值，其中 k 是回合數（即 `Bob()` 的呼叫次數）。則，你在該子任務的分數將乘以以下數值：

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

特別地，如果你在 $3n$ 回合內解決問題，你將獲得該子任務的滿分。使用超過 $12n$ 回合將導致該子任務獲得 0 分（顯示為「輸出錯誤」）。

範例評分程式

範例評分程式的輸入格式如下：

- 第 1 行：$m\ e$
- 第 $2+i$ 行（$0 \le i \le e-1$）：$u[i]\ v[i]$
- 第 $2+e$ 行：n
- 第 $3+e$ 行：$p[0]\ p[1]\ \ldots\ p[n-1]$

範例評分程式的輸出格式如下：

- 第 1 行：最終排列 p
- 第 2 行：`Alice()` 的回傳值
- 第 3 行：最終排列的實際分數
- 第 4 行：回合數