



## 最長旅程 (Longest Trip)

IOI 2023主辦單位遇到一大麻煩!他們忘了規劃幾天之後到 Ópusztaszer 的旅行。但也許還沒有太晚…

在Ópusztaszer有  $N$  座地標，其編號由  $0$  到  $N - 1$ 。在這些地標中，有些兩兩成對的地標之間有雙向連通的 **道路** 連接。每一對地標最多有一條道路連接。主辦單位 **不知道** 那些地標有道路連接。

假設每 3 座相異的地標，在它們之間至少有  $\delta$  條道路相連，我們稱此 Ópusztaszer 道路網絡的 **密度** 至少  $\delta$ 。換句話說，每 3 座地標  $(u, v, w)$ ，滿足  $0 \leq u < v < w < N$ ，在  $(u, v)$ 、 $(v, w)$  及  $(u, w)$  的地標對中，至少有  $\delta$  對有道路相連。

主辦單位 **知道** 一正整數  $D$  滿足道路網絡的密度至少是  $D$ 。注意  $D$  的值不能大於 3。

主辦單位可以透過 **呼叫** Ópusztaszer 的總機來收集關於特定地標間的道路連接資訊。在每一次呼叫，須指定兩個非空的地標陣列:  $[A[0], \dots, A[P - 1]]$  及  $[B[0], \dots, B[R - 1]]$ 。其中地標須兩兩相異，也就是，

- 對每一對  $i$  和  $j$ ， $0 \leq i < j < P$ ， $A[i] \neq A[j]$ ；
- 對每一對  $i$  和  $j$ ， $0 \leq i < j < R$ ， $B[i] \neq B[j]$ ；
- 對每一對  $i$  和  $j$ ， $0 \leq i < P$  及  $0 \leq j < R$ ， $A[i] \neq B[j]$ 。

每一次呼叫，總機回報是否有一道路連接  $A$  陣列中的地標至  $B$  陣列中的地標。準確地說，只要存在某對  $i$  和  $j$  ( $0 \leq i < P$ ) 和 ( $0 \leq j < R$ )，滿足地標  $A[i]$  及地標  $B[j]$  間有道路連接，則總機回傳 `true`。否則回傳 `false`。

一個長度為  $l$  的 **旅程** 是指由一相異地標編號的序列  $t[0], t[1], \dots, t[l - 1]$  所構成的旅程，其中對任一整數  $i$  ( $0 \leq i \leq l - 2$ )，編號為  $t[i]$  的地標和編號為  $t[i + 1]$  的地標間有道路相連。如果不存在長度大於  $l$  的旅程，則稱一個長度為  $l$  的 **旅程** 為 **最長旅程**。

你的任務是透過呼叫總機，協助主辦單位找出在 Ópusztaszer 的最長旅程。

## 實作細節 (Implementation Details)

你須實作下列程序：

```
int[] longest_trip(int N, int D)
```

- $N$ : 村莊 Ópusztaszer 的地標個數。
- $D$ : 道路網絡的最小密度。
- 這程序應回傳一陣列  $t = [t[0], t[1], \dots, t[l - 1]]$  代表某條最長旅程。

- 對每一筆測資，這程序可被呼叫 **多次**。

上述程序可呼叫下列程序若干次：

```
bool are_connected(int[] A, int[] B)
```

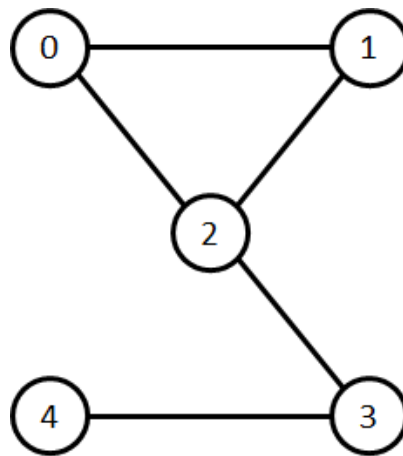
- $A$ ：一非空的陣列，表示相異的地標。
- $B$ ：一非空的陣列，表示相異的地標。
- $A$  和  $B$  沒有交集。
- 若  $A$  中有地標和  $B$  中的地標間有道路連接，則此程序回傳 `true`。否則回傳 `false`。
- 每次呼叫 `longest_trip` 時，此程序最多被呼叫 32 640 次，總共最多呼叫 150 000 次。
- 傳送陣列  $A$  和  $B$  給此程序的累計總長度不能超過 1 500 000。

評分程式是 **not adaptive**，即每次提交將以相同的測資評分。也就是在每一筆測資， $N$  和  $D$  的值以及由道路相連的地標對在每次呼叫 `longest_trip` 前對已先固定。

## 範例 (Examples)

### 範例一 (Example 1)

考慮一情境  $N = 5, D = 1$ ，以及道路連接的情況如下圖：



程序 `longest_trip` 以下列方式被呼叫：

```
longest_trip(5, 1)
```

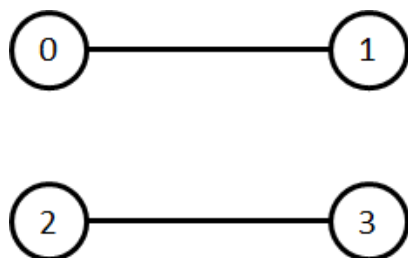
此程序呼叫 `are_connected` 若干次如下：

Call	Pairs connected by a road	Return value
<code>are_connected([0], [1, 2, 4, 3])</code>	(0,1) and (0,2)	true
<code>are_connected([2], [0])</code>	(2,0)	true
<code>are_connected([2], [3])</code>	(2,3)	true
<code>are_connected([1, 0], [4, 3])</code>	none	false

在第4次呼叫之後，結果 (1,4), (0,4), (1,3) 及 (0,3) 中 沒有任一對有道路相連。因道路網路的密度至少  $D = 1$ ，可以看出由三座地標 (0,3,4)中，(3,4) 這對地標間必有道路相連。同理，地標 0 及地標 1 必定相連。

此時，可以得知  $t = [1,0,2,3,4]$  是一長度為 5 的旅程，而且不存在長度超過 5 的旅程。因此，程序 `longest_trip` 可回傳 `[1,0,2,3,4]`。

考慮另一情境  $N = 4, D = 1$ ，以及地標之間道路連接方式如下圖：



程序 `longest_trip` 以下列方式被呼叫：

```
longest_trip(4, 1)
```

在此情境，最長旅程的長度為 2。因此在呼叫 `are_connected` 若干次後，程序 `longest_trip` 可回傳下列四者之一 `[0,1]`, `[1,0]`, `[2,3]`, `[3,2]`。

## 範例二 (Example 2)

子任務 0 包含一額外測資，其中地標個數  $N = 256$ 。此測資被包含在一附加檔案包 (attachment package)，此檔案包可在競賽系統下載。

## 限制 (Constraints)

- $3 \leq N \leq 256$
- 在每一筆測資，所有呼叫 `longest_trip` 的參數  $N$  總和不超過 1024。
- $1 \leq D \leq 3$

## 子任務 (Subtasks)

1. (5 points)  $D = 3$
2. (10 points)  $D = 2$
3. (25 points)  $D = 1$ 。令  $l^*$  表示最長旅程的長度。程序 `longest_trip` 不需回傳長度為  $l^*$  的旅程。而需回傳長度至少為  $\left\lceil \frac{l^*}{2} \right\rceil$  的旅程。
4. (60 points)  $D = 1$

在子任務 4 你的得分將由執行一次程序 `longest_trip` 時呼叫程序 `are_connected` 的次數來決定。在此子任務裡的每一筆測資，令  $q$  表示在所有執行程序 `longest_trip` 中最大的呼叫次數。你在此子任務的得分計算方式如下表：

Condition	Points
$2750 < q \leq 32640$	20
$550 < q \leq 2750$	30
$400 < q \leq 550$	45
$q \leq 400$	60

在任一測資，若呼叫程序 `are_connected` 不符合實作細節的限制，或程序 `longest_trip` 回傳的陣列不正確，則你在此子任務將得 0 分。

## 範例評分程式 (Sample Grader)

令  $C$  表示情境的個數，也就是呼叫 `longest_trip` 的次數。範例評分程式以下列格式讀取輸入：

- line 1:  $C$

$C$  種情境的描述如下。

範例評分程式針對每一種情境描述以下列格式讀取輸入：

- line 1:  $N \ D$
- line  $1 + i$  ( $1 \leq i < N$ ):  $U_i[0] \ U_i[1] \ \dots \ U_i[i - 1]$

在此，每一  $U_i$  ( $1 \leq i < N$ ) 是一大小為  $i$  的陣列，用以敘述哪一對地標有道路連接。對每一  $i$  和  $j$  ( $1 \leq i < N, 0 \leq j < i$ ):

- 若地標  $j$  和地標  $i$  有道路連接，則  $U_i[j]$  的值為 1；
- 若地標  $j$  和地標  $i$  沒有道路連接，則  $U_i[j]$  的值為 0。

在每一情境呼叫程序 `longest_trip` 前，範例評分程式將檢查道路網路密度是否至少為  $D$ 。若此條件不符合，將輸出 `Insufficient Density` 的訊息並停止評分程式。

若範例評分程式偵測到違反協定，將輸出 `Protocol Violation: <MSG>`，其中 `<MSG>` 將為下列錯誤訊息之一：

- `invalid array`: 在呼叫程序 `are_connected` 時，陣列  $A$  和  $B$  至少有一陣列:
  - 為空陣列，或
  - 包含不是  $0$  到  $N - 1$  的整數，或
  - 包含重複的數字。
- `non-disjoint arrays`: 在呼叫程序 `are_connected` 時，陣列  $A$  和  $B$  有交集
- `too many calls`: 在執行程序 `longest_trip` 時，呼叫程序 `are_connected` 的次數超過 32 640，或總次數超過 150 000 次。
- `too many elements`: 所有傳送到程序 `are_connected` 的地標總數超過 1 500 000。

否則，令程序 `longest_trip` 在一情境下回傳的陣列元素為:  $t[0], t[1], \dots, t[l - 1]$  ( $l \geq 0$ )。針對此情境以下列格式輸出三列:

- line 1:  $l$
- line 2:  $t[0] \ t[1] \ \dots \ t[l - 1]$
- line 3: 此情境下呼叫程序 `are_connected` 的次數。

最後輸出:

- line  $1 + 3 \cdot C$ : 在所有執行程序 `longest_trip` 中，呼叫程序 `are_connected` 的最多次數。