

最長旅程

IOI 2023 的組織者們遇到了大麻煩！他們忘記了為即將到來的一天前往 Ópusztaszer 去計劃旅程。但現在也許還不算太晚...

Ópusztaszer 有 N 個地標，從 0 到 $N - 1$ 編號。當中有一些地標之間會有 **雙向的道路** 連接。每一對地標之間最多只有一條道路連接。但組織者們並不知道哪些地標之間有道路連接。

如果每 3 個不同的地標之間至少都有 δ 條道路，我們稱 Ópusztaszer 的道路網絡的**密度**至少為 δ 。換句話說，對於每個地標三元組 (u, v, w) ，其中 $0 \leq u < v < w < N$ ，在地標對 (u, v) , (v, w) 和 (u, w) 中至少有 δ 對地標有道路連接。

組織者們知道一個正整數 D ，使得道路網絡的密度至少為 D 。注意， D 的值不能大於 3。

組織者們可以通過打電話給 Ópusztaszer 的調度員來收集有關某些地標之間道路連接的信息。在每次通話中，必須指定兩個非空的地標數組 $[A[0], \dots, A[P - 1]]$ 和 $[B[0], \dots, B[R - 1]]$ 。這些地標必須兩兩不同，即，

- 對於每個 i 和 j ，滿足 $0 \leq i < j < P$ ， $A[i] \neq A[j]$ ；
- 對於每個 i 和 j ，滿足 $0 \leq i < j < R$ ， $B[i] \neq B[j]$ ；
- 對於每個 i 和 j ，滿足 $0 \leq i < P$ 和 $0 \leq j < R$ ， $A[i] \neq B[j]$ 。

對於每次通話，調度員報告是否存在一條道路連接 A 中的一個地標和 B 中的一個地標。更準確地說，調度員會迭代所有地標對 i 和 j ，滿足 $0 \leq i < P$ 和 $0 \leq j < R$ 。如果對於其中任何一個地標對 $A[i]$ 和 $B[j]$ 有道路連接，則調度程序返回 “true”。否則，調度員返回 “false”。

一個長度為 l 的**旅程**是一個由不同地標 $t[0], t[1], \dots, t[l - 1]$ 組成的序列，其中對於每個 i ， $0 \leq i \leq l - 2$ ，地標 $t[i]$ 和地標 $t[i + 1]$ 由一條道路相連。如果不存在長度至少為 $l + 1$ 的旅程，則長度為 l 的旅程稱為**最長旅程**。

你的任務是通過向調度員打電話來幫助組織者們找到 Ópusztaszer 的最長旅程。

實現細節

你應該實現以下過程：

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

- N ：Ópusztaszer 的地標數量。

- D ：道路網絡的保證最小密度。
- 這個過程應該返回一個數組 $t = [t[0], t[1], \dots, t[l - 1]]$ ，表示一個最長旅程。
- 在每個測試用例中，這個過程可能被**多次**調用。

上述過程可以調用以下過程：

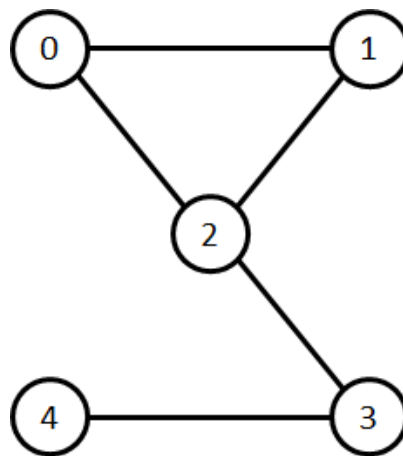
```
bool are_connected(int[] A, int[] B)
```

- A ：一個非空的不同地標數組。
- B ：一個非空的不同地標數組。
- A 和 B 應該是不相交的。
- 如果存在一個地標來自 A ，一個地標來自 B ，並且它們之間有一條道路相連，則這個過程返回 true。否則，返回 false。
- 在每次調用 `longest_trip` 中，這個過程最多可以被調用 32 640 次，總共最多可以被調用 150 000 次。
- 在所有調用中，傳遞給這個過程的數組 A 和 B 的總長度不能超過 1 500 000。

範例

範例 1

考慮一個情境，其中 $N = 5$ ， $D = 1$ ，道路連接如下圖所示：



以以下方式呼叫 `longest_trip` 函式：

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

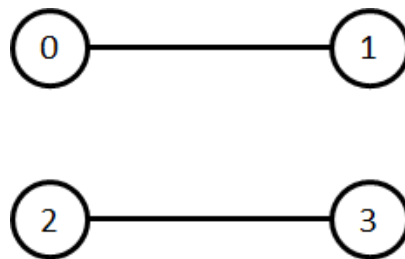
函式可能會呼叫 `are_connected` 函式如下。

呼叫	由道路連接的對組	回傳值
<code>are_connected([0], [1, 2, 4, 3])</code>	(0,1) 和 (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>	無	false

在第四次呼叫之後，發現 (1,4)、(0,4)、(1,3) 和 (0,3) 這四個對組都沒有被道路連接。由於網路的密度至少為 $D = 1$ ，我們可以看到從三元組 (0,3,4) 中，對組 (3,4) 必須被道路連接。類似地，地標 0 和 1 必須被連接。

此時，可以得出 $t = [1, 0, 2, 3, 4]$ 是一條長度為 5 的旅程，且不存在長度大於 5 的旅程。因此，函式 `longest_trip` 可以回傳 `[1, 0, 2, 3, 4]`。

考慮另一個情境，其中 $N = 4$ ， $D = 1$ ，地標之間的道路如下圖所示：



子程序會以下方式去調用 `longest_trip` 函式：

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

在這個情境中，最長旅程的長度為 2。因此，在對 `are_connected` 函式的幾次呼叫之後，函式 `longest_trip` 可以回傳其中一個 `[0, 1]`、`[1, 0]`、`[2, 3]` 或 `[3, 2]`。

範例 2

子任務 0 包含一個額外的範例測試案例，其中 $N = 256$ 個地標。這個測試案例包含在您可以從競賽系統下載的附件包中。

限制

- $3 \leq N \leq 256$
- 所有對 `longest_trip` 的呼叫中的 N 的總和不超過 1024。
- $1 \leq D \leq 3$

子任務

1. (5 分) $D = 3$
2. (10 分) $D = 2$
3. (25 分) $D = 1$ 。令 l^* 表示最長旅程的長度。函式 `longest_trip` 不必回傳長度為 l^* 的旅程。相反，它應該回傳長度至少為 $\left\lceil \frac{l^*}{2} \right\rceil$ 的旅程。
4. (60 分) $D = 1$

如果在任何測試案例中，對函式 `are_connected` 的呼叫不符合實作細節中描述的限制，或者 `longest_trip` 回傳的陣列不正確，則該子任務的您的解答得分將為 0。

在子任務 4 中，您的得分基於在單次呼叫 `longest_trip` 中對函式 `are_connected` 的呼叫次數。令 q 為在子任務的所有測試案例中，對 `longest_trip` 的每次呼叫中的呼叫數的最大值。您在此子任務中的得分根據以下表格計算：

條件	分數
$2\,750 < q \leq 32\,640$	20
$550 < q \leq 2\,750$	30
$400 < q \leq 550$	45
$q \leq 400$	60

範例評分程式

令 C 表示情境的數量，也就是對 `longest_trip` 的呼叫次數。範例評分程式以下格式讀取輸入：

- 第 1 行： C

接下來是 C 個情境的描述。

範例評分程式以下格式讀取每個情境的描述：

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

在這裡，每個 U_i ($1 \leq i < N$) 都是一個大小為 i 的陣列，描述了哪些地標之間有道路相連。對於每個 i 和 j ，滿足 $1 \leq i < N$ 且 $0 \leq j < i$ ：

- 如果地標 j 和 i 之間有道路相連，則 $U_i[j]$ 的值應為 1；
- 如果地標 j 和 i 之間沒有道路相連，則 $U_i[j]$ 的值應為 0。

在每個情境中，在呼叫 `longest_trip` 之前，範例評分程式會檢查道路網絡的密度是否至少為 D 。如果不滿足此條件，它會輸出訊息 `Insufficient Density` 並終止。

如果範例評分程式檢測到協議違規，則範例評分程式的輸出為 Protocol Violation: <MSG>，其中 <MSG> 是以下錯誤訊息之一：

- invalid array: 在呼叫 are_connected 時，至少有一個陣列 A 和 B
 - 是空的，或
 - 包含一個不是介於 0 和 $N - 1$ （包含）之間的整數，或
 - 包含至少兩次相同的元素。
- non-disjoint arrays: 在呼叫 are_connected 時，陣列 A 和 B 不是不相交的。
- too many calls: 對 are_connected 的呼叫次數超過了當前對 longest_trip 的呼叫中的 32 640 次，或者總共超過了 150 000 次。
- too many elements: 在所有呼叫中，傳遞給 are_connected 的地標總數超過了 1 500 000。

否則，令某個情境中 longest_trip 返回的陣列的元素為 $t[0], t[1], \dots, t[l - 1]$ ，其中 l 是非負整數。範例評分程式以以下格式為該情境打印三行：

- 第 1 行: l
- 第 2 行: $t[0] \ t[1] \ \dots \ t[l - 1]$
- 第 3 行: 在該情境中對 are_connected 的呼叫次數

最後，範例評分程式打印：

- 第 $1 + 3 \cdot C$ 行: 所有對 longest_trip 的呼叫中對 are_connected 的最大呼叫次數