



千島群島 (Thousands Islands)

千島群島 (Thousands Islands) 指的是爪哇海 (Java Sea) 上的一群美麗小島。它包含了 N 座島嶼，其編號為 0 至 $N - 1$ 。

有 M 條獨木舟航行於島嶼之間，其編號為 0 至 $M - 1$ 。對所有的 i ， $0 \leq i \leq M - 1$ ，編號為 i 的獨木舟可以停泊於島嶼 $U[i]$ 或 $V[i]$ ，並且航行於島嶼 $U[i]$ 與島嶼 $V[i]$ 之間。具體來說，當這條獨木舟停泊於島嶼 $U[i]$ 時，可以划著它從島嶼 $U[i]$ 航行至島嶼 $V[i]$ ，之後該獨木舟便停泊於島嶼 $V[i]$ 。類似地，當這條獨木舟停泊於島嶼 $V[i]$ 時，可以划著它從島嶼 $V[i]$ 航行至島嶼 $U[i]$ ，之後該獨木舟便停泊於島嶼 $U[i]$ 。一開始的時候，這條獨木舟停泊於島嶼 $U[i]$ 。此外，可能有多條獨木舟被用來航行於同一對島嶼之間；也可能同時有多條獨木舟停泊於同一座島嶼。

為了安全起見，一條獨木舟每次航行後必須要進行保養。因此，你無法划著它連續航行兩次。也就是說，使用了獨木舟 i 以後，再次使用它之前必須要使用另一條獨木舟。

Bu Dengklek 希望規劃一趟航行於某些島嶼之間的旅程。她的旅程是「合法的」若且唯若以下的條件被滿足：

- 旅程的起點與終點皆為島嶼 0 。
- 她造訪了至少一座有別於島嶼 0 的島嶼。
- 當旅程結束後，所有的獨木舟將停泊於它們一開始停泊的島嶼。也就是說，在旅程結束後，對於所有 i ， $0 \leq i \leq M - 1$ ，獨木舟 i 必須要停泊於島嶼 $U[i]$ 。

請幫助 Bu Dengklek 找到一條航行至多 $2\,000\,000$ 次的合法旅程，或判斷合法的旅程不存在。可以證明，在本題的條件限制之下（詳見限制章節），若存在合法的旅程，那麼必定存在航行次數至多為 $2\,000\,000$ 次的合法旅程。

實作細節 (Implementation Details)

你應該實作以下程序：

```
union(bool, int[]) find_journey(int N, int M, int[] U, int[] V)
```

- N : 島嶼的總數。
- M : 獨木舟的總數。
- U, V : 長度為 M 的陣列，用以描述獨木舟。
- 此程序應回傳一個布林值或是一個整數陣列。
 - 若不存在任何合法的旅程，該程序應回傳 `false`。
 - 若存在合法的旅程，則你有兩種選擇：

- 要拿到滿分，該程序應回傳一個長度至多為 2 000 000 的整數陣列，用以描述一趟合法的旅程。更精確地說，陣列中的整數依序代表旅程中使用的獨木舟編號。
 - 要拿到部分分數，你的程序應回傳 true、回傳一個長度超過 2 000 000 的整數陣列、或是回傳的整數陣列描述了一趟不合法的旅程 (詳見子任務章節)。
- 此程序會被呼叫一次。

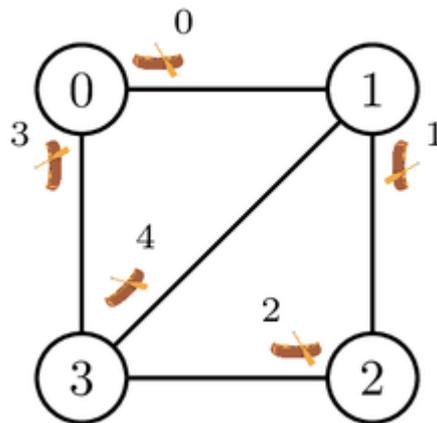
範例 (Examples)

範例 1 (Example 1)

考慮以下呼叫：

```
find_journey(4, 5, [0, 1, 2, 0, 3], [1, 2, 3, 3, 1])
```

範例中的島嶼和獨木舟以下圖表示：



一趟可能的合法旅程由下所述：Bu Dengklek 首先依序使用編號為 0、1、2 以及 4 的獨木舟航行。航行後她將抵達島嶼 1。接著，Bu Dengklek 再次划著編號 0 的獨木舟 (該獨木舟目前停泊於島嶼 1，且前一次航行並非使用獨木舟 0)。使用獨木舟 0 航行後，Bu Dengklek 抵達了島嶼 0。但是，獨木舟 1、2 以及 4 並沒有停泊於它們初始所在的島嶼。Bu Dengklek 延續了她的旅程，依序使用了獨木舟 3、2、1、4 以及 3 來航行。現在 Bu Dengklek 回到了島嶼 0，而且所有的獨木舟也都停泊在它們一開始所在的島嶼。

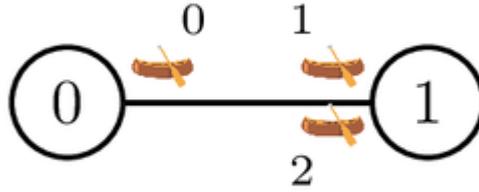
因此，回傳值 [0, 1, 2, 4, 0, 3, 2, 1, 4, 3] 描述了一趟合法的旅程。

範例 2 (Example 2)

考慮以下呼叫：

```
find_journey(2, 3, [0, 1, 1], [1, 0, 0])
```

範例中的島嶼和獨木舟以下圖表示：



Bu Dengklek 一開始只能划獨木舟 0，航行後她能選擇划獨木舟 1 或獨木舟 2。請注意，此時她不能連續划獨木舟 0。無論她選擇哪一條獨木舟，她都會回到島嶼 0。但是，Bu Dengklek 再也無法繼續她的旅程了，因為這個時候唯一停泊在島嶼 0 的獨木舟是她才剛航行過的。此外，目前為止的旅程也並非合法，因為仍有獨木舟停泊在與旅程開始時不同的島嶼。

由於不存在合法的旅程，因此該程序應回傳 false。

限制 (Constraints)

- $2 \leq N \leq 100\,000$
- $1 \leq M \leq 200\,000$
- $0 \leq U[i] \leq N - 1$ 且 $0 \leq V[i] \leq N - 1$ (對所有 i , $0 \leq i \leq M - 1$)
- $U[i] \neq V[i]$ (對所有 i , $0 \leq i \leq M - 1$)

子任務 (Subtasks)

1. (5 points) $N = 2$
2. (5 points) $N \leq 400$ 。對於所有不同的島嶼配對 x 與 y ($0 \leq x < y \leq N - 1$)，恰好有兩條獨木舟得以航行於該兩島之間。其中一條初始時停泊於島嶼 x 、另一條初始時停泊於島嶼 y 。
3. (21 points) $N \leq 1000$ ； M 為偶數；對於所有的「偶數」 i 且 $0 \leq i \leq M - 1$ ，獨木舟 i 與獨木舟 $i + 1$ 皆可以航行於島嶼 $U[i]$ 與 $V[i]$ 之間。獨木舟 i 初始時停泊於島嶼 $U[i]$ 、獨木舟 $i + 1$ 初始時停泊於島嶼 $V[i]$ 。也就是說， $U[i] = V[i + 1]$ 而且 $V[i] = U[i + 1]$ 。
4. (24 points) $N \leq 1000$ ； M 為偶數；對於所有的「偶數」 i 且 $0 \leq i \leq M - 1$ ，獨木舟 i 與獨木舟 $i + 1$ 皆可以航行於島嶼 $U[i]$ 與 $V[i]$ 之間。這兩條獨木舟初始時皆停泊於島嶼 $U[i]$ 。也就是說， $U[i] = U[i + 1]$ 而且 $V[i] = V[i + 1]$ 。
5. (45 points) 無額外限制。

對於所有存在一趟合法旅程的測試資料，你的答案：

- 將得到滿分 —— 若回傳一趟合法的旅程；
- 將得到 35% 的分數 —— 若回傳 true、回傳一個長度超過 2 000 000 的整數陣列、或是回傳的整數陣列描述了一趟不合法的旅程；
- 在其他情形下，將得到 0 分。

對於所有不存在一趟合法旅程的測試資料，你的答案：

- 將得到滿分 —— 若回傳 `false` ；
- 在其他情形下，將得到 0 分。

請注意，每一個子任務的最終得分，為該子任務中所有測試資料得分的最小值。

範例評分程式 (Sample Grader)

範例評分程式以下列格式讀取輸入：

- line 1: $N M$
- line $2 + i$ ($0 \leq i \leq M - 1$): $U[i] V[i]$

範例評分程式將你的答案以下列格式輸出：

- 若 `find_journey` 回傳了一個布林型態的值 (`bool`):
 - line 1: 0
 - line 2: 若 `find_journey` 回傳了 `false` 則輸出 0，否則輸出 1。
- 若 `find_journey` 回傳了一個整數陣列 (`int[]`)，令陣列中的數字依序為 $c[0], c[1], \dots, c[k - 1]$ 。
則範例評分程式將輸出：
 - line 1: 1
 - line 2: k
 - line 3: $c[0] c[1] \dots c[k - 1]$