



Thousands Islands (千島群島)

千島群島是位於爪哇海（Java Sea）的一群美麗的小島嶼串連而成。它包含了 N 個島嶼，編號為從 0 到 $N - 1$ 。

有 M 艘獨木舟，編號為從 0 到 $M - 1$ ，用於島嶼之間的航行。對於每個滿足 $0 \leq i \leq M - 1$ 的 i ，獨木舟 i 可以停靠於島嶼 $U[i]$ 或 $V[i]$ ，且能在島嶼 $U[i]$ 和 $V[i]$ 之間航行。具體來說，當獨木舟停靠於島嶼 $U[i]$ ，它可以用於從島嶼 $U[i]$ 航行到島嶼 $V[i]$ ，然後停靠於島嶼 $V[i]$ 。同樣，當獨木舟停靠於島嶼 $V[i]$ ，它可以用於從島嶼 $V[i]$ 航行到島嶼 $U[i]$ ，然後停靠於島嶼 $U[i]$ 。一開始，獨木舟停靠於島嶼 $U[i]$ 。可存在多於一艘獨木舟能在同一對島嶼之間航行。亦可存在多於一艘獨木舟停靠在同一個島嶼。

為安全起見，一艘獨木舟每次被使用後都需要維護，即是禁止同一艘獨木舟連續航行兩次。換言之，獨木舟 i 使用後，必需在其它獨木舟使用後才能再次使用。

Bu Dengklek 希望計劃一個旅程來遊玩一些島嶼。要使旅程**合法**，必需滿足下列的條件。

- 旅程的開始和結束都是位於島嶼 0。
- 到訪至少一個除島嶼 0 外的其它島嶼。
- 在旅程結束後，所有獨木舟需停靠於旅程開始時的位置。即是，對於每個滿足 $0 \leq i \leq M - 1$ 的 i ，獨木舟 i 必需停靠於島嶼 $U[i]$ 。

請幫助 Bu Dengklek 找出任何不多於 2 000 000 次航行的合法的旅程，或判定不存在合法的旅程。可以證明，在此任務的限制條件下（詳見限制條件），如果存在一個合法旅程，則一定存在不多於 2 000 000 次航行的合法的旅程。

編程實現細節

你需要實現以下的子程序：

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

- N : 島嶼的數量。
- M : 獨木舟的數量。
- U, V : 長度為 M 的兩個數組。
- 這個子程序應該返回一個 Boolean 或者一個整數數組。
 - 如果不存在合法的旅程，這個子程序應該返回 false。
 - 如果存在合法的旅程，則你可以有兩個做法：
 - 要獲得滿分，該過程應返回一個最多包含 2 000 000 整數的數組，表示有效旅程。更準確地說，這個數組的元素應該是旅途中使用的獨木舟的數量（按使用順序）。

- 要獲得部分分數，程序應返回 true, 一個超過 2 000 000 整數的數組，或不描述有效旅程的整數數組。（有關詳細信息，請參閱子任務部分。）
- 這個子程序只會被調用剛好一次。

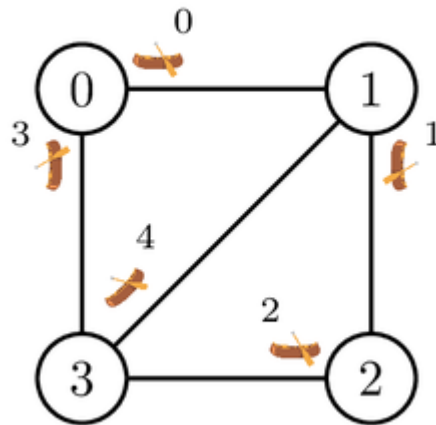
樣例

樣例 1

考慮以下調用：

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

島嶼及獨木舟如下圖所示：



一程合法的旅程如下所述：Bu Dengklek 先分別使用獨木舟 0、1、2、4 航行。結果，她移動到島嶼 1。然後，Bu Dengklek 再次使用停靠於島嶼 1 的獨木舟 0（獨木舟 0 已不在維護狀態）。使用獨木舟 0 航行後，Bu Dengklek 移到島嶼 0。然而，獨木舟 1、2、4 未停靠於開始旅程前的島嶼。Bu Dengklek 繼續她的旅程，使用獨木舟 3、2、1、4 航行，然後再次使用獨木舟 3。Bu Dengklek 回到島嶼 0，並且所有獨木舟停靠於開始旅程前的島嶼。

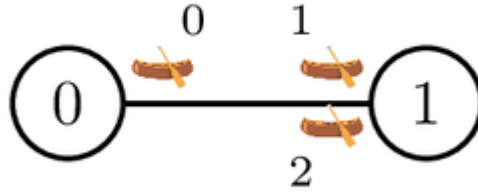
所以，`[0, 1, 2, 4, 0, 3, 2, 1, 4, 3]` 是子程式的一個合法的返回結果。

樣例 2

考慮以下調用：

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

島嶼及獨木舟如下圖所示



Bu Dengklek 一開始只可以使用獨木舟 0 航行，然後她只能使用獨木舟 1 或 2。請注意，她不能連續兩次使用獨木舟 0 航行。使用獨木舟 1 或 2 後都會返回島嶼 0。然而，獨木舟不是停靠於開始旅程前的島嶼，Bu Dengklek 無法使用任何獨木舟，因為現在停靠於島嶼 0 的唯一獨木舟是她剛使用過的。因此，不存在合法的旅程，子程式應該返回 false。

限制條件

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

子任務

1. (5 分) $N = 2$
2. (5 分) $N \leq 400$. 對於每對不同的島嶼 x 和 y ($0 \leq x < y \leq N - 1$)，正好有兩艘獨木舟可以用來在它們之間航行。一艘停靠於島嶼 x ，另一艘停靠於島嶼 y 。
3. (21 分) $N \leq 1000$ ， M 是偶數，對於每個滿足 $0 \leq i \leq M - 1$ 的偶數 i ，獨木舟 i 和 $i + 1$ 可以使用於島嶼 $U[i]$ 和 $V[i]$ 之間的航行。獨木舟 i 初始停靠於島嶼 $U[i]$ ，獨木舟 $i + 1$ 初始停靠於島嶼 $V[i]$ 。即是， $U[i] = V[i + 1]$ 和 $V[i] = U[i + 1]$ 。
4. (24 分) $N \leq 1000$ ， M 是偶數，對於每個滿足 $0 \leq i \leq M - 1$ 的偶數 i ，獨木舟 i 和 $i + 1$ 可以使用於島嶼 $U[i]$ 和 $V[i]$ 之間的航行。兩艘獨木舟 i 初始都是停靠於島嶼 $U[i]$ 。即是， $U[i] = U[i + 1]$ 和 $V[i] = V[i + 1]$ 。
5. (45 分) 沒有其它附加條件。

對於每個存在合法旅程的測試樣例，你的解決方案為：

- 如果返回的是一個合法的旅程，可取得滿分，
- 如果返回 true，或者返回一個超過 2 000 000 個整數的數組，或者返回一個不合法的旅程的數組，都可取得 35% 分數，
- 否則，分數為 0。

對於每個不存在合法旅程的測試用例，你的方案：

- 如果返回 false，可取得滿分；

- 否則，分數為 0。

請注意，每個子任務的最終分數是子任務中每個測試用例的最低分數。

樣例評分器

樣例評分器所讀入的輸入資料格式下：

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

樣例評分器將會以以下的格式來輸出你的答案：

- 如果 `find_journey` 返回一個 `bool`:
 - 第 1 行: 0
 - 第 2 行: 0 if `find_journey` returns false, or 1 otherwise.
- 如果 `find_journey` 返回一個 `int[]`, 使用 $c[0], c[1], \dots, c[k - 1]$ 表示這個數組。樣例評分器輸出：
 - 第 1 行: 1
 - 第 2 行: k
 - 第 3 行: $c[0] c[1] \dots c[k - 1]$