

遊戲 (Game)

在探索到 n 個星球後，法老們將星球由 0 編號至 $n - 1$ ，並利用「單向傳送點」建造一個星球間的運輸系統。每個傳送點有一個起點星球與一個終點星球。當一個旅遊者使用一傳送點時，將會由該傳送點的起點星球被傳送至終點星球。注意一個傳送點的起點星球與終點星球可能相同。一個以 u 為起點星球與 v 為終點星球的傳送點被表示為 (u, v) 。

為了提升傳送系統的使用普及率，法老們發明了一個遊戲讓旅遊者們在利用傳送系統旅遊時玩。每位旅遊者可在任一星球開始此遊戲。星球 $0, 1, \dots, k - 1$ ($k \leq n$) 被稱為「特殊星球」。每當一位旅遊者進入一個特殊星球時，該旅遊者會獲得一個戳記。

現在，對每個 i ($0 \leq i \leq k - 2$) 都有一個傳送點 $(i, i + 1)$ 。此 $k - 1$ 個傳送點被稱為「起始傳送點」。

新的傳送點是一個一個被加入的。每當一個新的傳送點被加入，旅遊者就有可能因此而能獲得無窮多個戳記。更精確地，當有一個星球的序列 $w[0], w[1], \dots, w[t]$ 滿足下列條件時，就會發生前述的情形：

- $1 \leq t$
- $0 \leq w[0] \leq k - 1$
- $w[t] = w[0]$
- 對每個 i ($0 \leq i \leq t - 1$)，有傳送點 $(w[i], w[i + 1])$ 。

注意，旅遊者可以使用起始傳送點以及當下已被加入的「任何」傳送點。

你的任務是幫助法老們確認，在每次加入一個傳送點後，是否有旅遊者能獲得無窮多個戳記。

實作細節 (Implementation details)

你應實作下列函式：

```
init(int n, int k)
```

- n : 星球數。
- k : 特殊星球數。
- 此函式將被呼叫恰好一次，該呼叫會在所有 `add_teleporter` 的呼叫之前。

```
int add_teleporter(int u, int v)
```

- u 和 v : 被加入的傳送點之起點星球以及終點星球。
- 此函式被呼叫至多 m 次 (m 的範圍請見限制 (Constraints))。

- 在傳送點 (u, v) 加入後，若旅遊者能獲得無窮多個戳記，則此函式應回傳 1。否則，此函式應回傳 0。
- 一旦此函式回傳 1，你的程式執行就會被終止。

範例 (Examples)

Example 1

考慮以下呼叫：

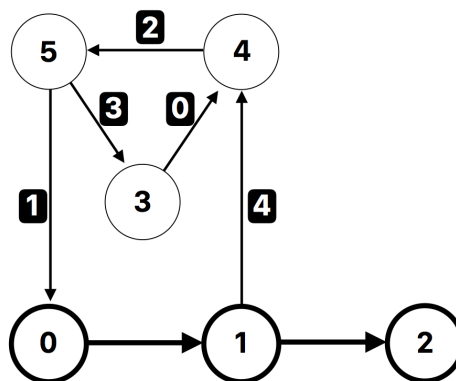
```
init(6, 3)
```

此範例中，共有 6 個星球，其中 3 個為特殊星球。星球 0、1 與 2 為特殊星球。起始傳送點為 $(0, 1)$ 與 $(1, 2)$ 。

假設評分程式呼叫：

- (0) `add_teleporter(3, 4)`: 你應回傳 0.
- (1) `add_teleporter(5, 0)`: 你應回傳 0.
- (2) `add_teleporter(4, 5)`: 你應回傳 0.
- (3) `add_teleporter(5, 3)`: 你應回傳 0.
- (4) `add_teleporter(1, 4)`: 此刻獲得無窮多個戳記是可能的。例如，旅遊者從星球 0 出發，依序前往 1, 4, 5, 0, 1, 4, 5, 0, ...。因此，你應回傳 1，而你的程式執行就會被終止。

下圖表示了這個範例。特殊星球與起始傳送點以粗線表示。被 `add_teleporter` 加入的傳送點依序以 0 到 4 標記。



Example 2

考慮以下呼叫：

```
init(4, 2)
```

此範例中，共有 4 個星球，其中 2 個為特殊星球。星球 0 與 1 為特殊星球。起始傳送點為 $(0, 1)$ 。

假設評分程式呼叫：

- `add_teleporter(1, 1)`：當加入傳送點 $(1, 1)$ 後，獲得無窮多個戳記是可能的。例如，旅遊者從星球 1 出發，利用傳送點 $(1, 1)$ 進入星球 1 無窮多次。因此，你應回傳 1，而你的程式執行就會被終止。

另一個範例輸入輸出可於附件包裹中取得。

限制 (Constraints)

- $1 \leq n \leq 300\,000$
- $1 \leq m \leq 500\,000$
- $1 \leq k \leq n$

對每個 `add_teleporter` 的呼叫：

- $0 \leq u \leq n - 1$ 且 $0 \leq v \leq n - 1$
- 在加入傳送點 (u, v) 前，不會有星球 u 至星球 v 的傳送點。

子任務 (Subtasks)

1. (2 points) $n = k, n \leq 100, m \leq 300$
2. (10 points) $n \leq 100, m \leq 300$
3. (18 points) $n \leq 1\,000, m \leq 5\,000$
4. (30 points) $n \leq 30\,000, m \leq 50\,000, k \leq 1\,000$
5. (40 points) 無額外限制

範例評分程式 (Sample grader)

此範例評分程式用以下格式讀取輸入：

- line 1: $n\ m\ k$
- line $2 + i$ ($0 \leq i \leq m - 1$): $u[i]\ v[i]$

此範例評分程式首先呼叫 `init`，然後依 $i = 0, 1, \dots, m - 1$ 的順序對 $u = u[i]$ 與 $v = v[i]$ 呼叫 `add_teleporter`。

它會輸出首次回傳 1 的 `add_teleporter` 呼叫之索引 (介於 0 與 $m - 1$ ，包含此二邊界值)；若所有 `add_teleporter` 呼叫皆回傳 0，則輸出 m 。

若某次 `add_teleporter` 呼叫回傳 0 與 1 以外的整數，此範例評分程式會輸出 -1 且你的程式會立即被終止。