



站點 (stations)

新加坡的互聯網骨幹網 (SIB) 是由 n 個站點組成，這些站點分配了從 0 到 $n - 1$ 的 **編號**。網絡內亦有 $n - 1$ 條雙向連線，其編號是從 0 到 $n - 2$ 。每條連線連接兩個不同的站點。通過單一條連線連接的兩個站稱為鄰居。

從站點 x 到站點 y 的路徑是一個由不同的站點 a_0, a_1, \dots, a_p 組成的序列，其中 $a_0 = x, a_p = y$ 且其內每一對連續的站點均為鄰居。從任何一個站點 x 到任何另一站點 y 都只存在 **唯一一條路徑**。

任何站點 x 都可以創建一個數據包（一條數據）並將其發送到任何其他站點 y （稱為數據包的 **目標站點**）。該數據包必須沿著唯一的一條路徑從 x 路由到 y ，如下說明：考慮當前保存有數據包的站點 z ，其目標站點為 y ($z \neq y$)。站點 z 需要：

1. 執行“路由程序”來確定 z 的哪個鄰居是在由 z 到 y 唯一路徑上，並且
2. 將該數據包轉發到該鄰居。

但是，站點的內存有限，不能將 SIB 內所有的連線資料都存起來給路由程序使用。

你的工作是要為 SIB 實現路由方案，該方案將包括以下兩個子程序。

- 第一個子程序是給定 n 的值，SIB 內連線的表列以及一個整數 $k \geq n - 1$ 作為輸入。你的程序要為每個站點分配一個介於 0 和 k 之間的獨一無二的整數 **標籤**。
- 第二個子程序是路由子程序，這個子程序在分配好站點的標籤後將會被部署到所有站點上。對於這個子程序，系統 **只提供** 以下的輸入：
 - s ，當前存放數據包的站點的 **標籤**，
 - t ，數據包目標站點的 **標籤** ($t \neq s$)，
 - c ，是 s 所有鄰居的 **標籤** 表列。

它應返回數據包應轉發到的 s 的鄰居的 **標籤**。

在其中一個子任務中，你的解決方案的得分數是取決於分配給任何站點的最大標籤的值（通常，越小越好）。

編程實作細節

你需要實現以下的子程序：

```
int[] label(int n, int k, int[] u, int[] v)
```

- n : SIB 內站點的數目。

- k : 標籤的最大值。
- u 及 v : 是大小為 $n - 1$ 的數組，它們表示著系統內的連線。對於每個 i ($0 \leq i \leq n - 2$)，連線 i 連接編號為 $u[i]$ 和 $v[i]$ 的站點。
- 此子程序應返回單個數組 L ，大小為 n 。對於每個 i ($0 \leq i \leq n - 1$) $L[i]$ 是分配給編號為 i 站點的標籤。數組 L 的所有元素必須是唯一的，且其範圍在 0 和 k 之間（含邊界）。

```
int find_next_station (int s, int t, int[ ] c)
```

- s : 持有數據包的站點的標籤。
- t : 數據包的目標站點的標籤。
- c : 一個數組，它是 s 的所有鄰居的標籤表列。數組 c 按升序排列著。
- 此子程序應返回數據包應轉發到的 s 的鄰居的標籤。

每個測試數據都涉及一個或多個獨立場景（即，不同的 SIB 描述）。對於有 r 個場景的測試數據，上述的子程序將被 **評分程式** 調用恰好兩次，如下所述:

在程式的第一次運行期間：

- label 子程序會被調用 r 次，
- 所返回的標籤資料會被評分程式存儲起來，並且
- find_next_station 不會在這時被調用。

在程式的第二次運行期間：

- find_next_station 可能被多次調用，
- 每次調用 find_next_station 時，輸入標籤是從第一次運行中以 **任意選擇** 場景的對 label 的調用時所產生的標籤，並且
- 這時不會調用 label。

特別地，在程序的第一次運行中，任何保存到靜態或全局變量的信息在調用 find_next_station 時均不可使用。

樣例

考慮以下的子程序調用：

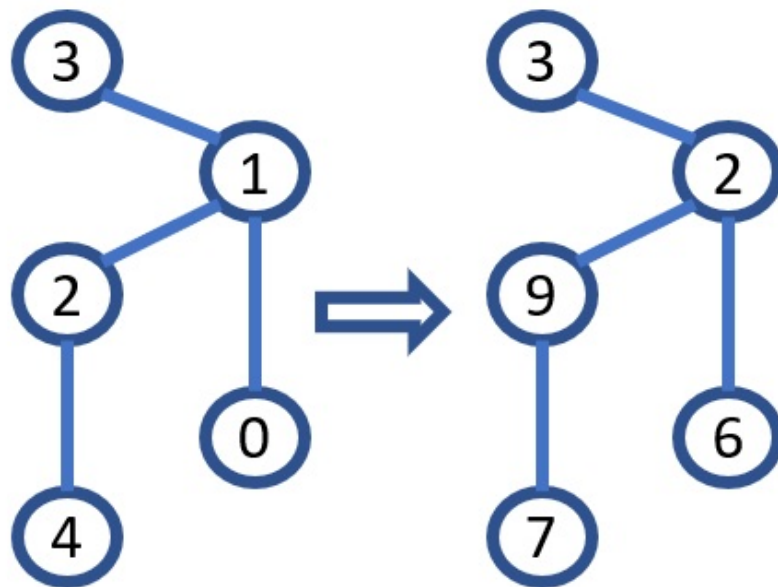
```
label(5, 10, [0, 1, 1, 2], [1, 2, 3, 4])
```

總共有 5 個站點和 4 條連線連接編號為 $(0, 1)$ ， $(1, 2)$ ， $(1, 3)$ 和 $(2, 4)$ 的站點對。每個標籤可以是從 0 到 $k = 10$ 的整數。

為了報告以下標籤：

編號	標籤
0	6
1	2
2	9
3	3
4	7

label 子程序應返回 [6, 2, 9, 3, 7]。在下圖中的數字，左側的子圖顯示站點的編號，右側的子圖顯示分配給每個站點的標籤。



假設標籤已如上所述的形式來分配，考慮以下調用：

```
find_next_station(9, 6, [2, 7])
```

這意味著持有數據包的站點的標籤為 9，目標站點的標籤為 6。到目標站點路徑上的站點標籤為 [9, 2, 6]。因此，子程序應返回 2，這是數據包應轉發到的站的標籤（該站點的原本編號為 1）。

考慮另一個可能的子程序調用：

```
find_next_station(2, 3, [3, 6, 9])
```

子程序應返回 3，因為標籤為 3 的目標站點是標籤為 2 的站的鄰居，因此應直接接收該數據包。

限制條件

- $1 \leq r \leq 10$

對於每一次調用 `label`:

- $2 \leq n \leq 1000$
- $k \geq n - 1$
- $0 \leq u[i], v[i] \leq n - 1$ (for all $0 \leq i \leq n - 2$)

對於所有調用 `find_next_station`, 其輸入資料是來自之前任意一次調用 `label` 的返回值。考慮它所生成的標籤:

- s 和 t 是兩個不同站點的標籤。
- c 是標籤 s 的所有鄰居並以升序排列的數列

在每個測試數據內, 若把所有場景加起來, 傳遞給過程 `find_next_station` 的所有數組 c 的長度總和不超過 100 000。

子任務

1. (5 分) $k = 1000$, 沒有任何站點會有多於 2 個鄰居。
2. (8 分) $k = 1000$, 連線把站點 i 連到站點 $i + 1$ 及 $\lfloor \frac{i}{2} \rfloor$ 。
3. (16 分) $k = 1\,000\,000$, 最多只有一個站點會有多於 2 個鄰居。
4. (10 分) $n \leq 8, k = 10^9$
5. (61 分) $k = 10^9$

在子任務 5 中, 你可以獲得部分分數。 假設 m 是所有情況下 `label` 返回的最大標籤值。 根據下表計算此子任務的分數:

最大標籤值	分數
$m \geq 10^9$	0
$2000 \leq m < 10^9$	$50 \cdot \log_{5 \cdot 10^5}(\frac{10^9}{m})$
$1000 < m < 2000$	50
$m \leq 1000$	61

樣例評分程式

樣例評分程式會以以下的格式輸入所需的數據:

- 第 1 行: r

隨後有 r 塊資料, 每個塊資料描述一個場景。 每個塊的格式如下:

- 第 1 行: $n \ k$
- 第 $2 + i$ ($0 \leq i \leq n - 2$) 行: $u[i] \ v[i]$
- 第 $1 + n$ 行: q 是調用 `find_next_station` 次數.
- 第 $2 + n + j$ ($0 \leq j \leq q - 1$) 行: $z[j] \ y[j] \ w[j]$: 調用 `find_next_station` 第 j 次時所

涉及的站點的 **編號**：站點 $z[j]$ 保存數據包，站點 $y[j]$ 是數據包的目標，而站點 $w[j]$ 是在從 $z[j]$ 到 $y[j]$ 的唯一路徑上跟隨 $z[j]$ 的站點。

樣例評分程式會以以下的格式把結果打印出來：

- 第 1 行: m

與輸入中的連續場景相對應的 r 塊如下。每個塊的格式如下：

- 第 $1 + j$ ($0 \leq j \leq q - 1$) 行：站點的 **編號**，其 **標籤**是在這個場景調用 `find_next_station` 的 j 次返回的值。

請注意，樣例評分程式的每次運行都調用 `label` 和 `find_next_station`。