

站點 (stations)

新加坡網際網路骨幹 (Singapore's Internet Backbone, SIB) 由 n 個站點構成，分別有 站點編號 0 至 $n - 1$ 。另外也有 $n - 1$ 條雙向的連結 (bidirectional link)，分別編號由 0 至 $n - 2$ 。每個連結接通兩個不同的站點。兩站點間由單一連結接通的稱為鄰居。

一條由站點 x 至站點 y 的路徑是一由不同站點 a_0, a_1, \dots, a_p 構成的序列，滿足 $a_0 = x, a_p = y$ ，且序列中連續二站點為鄰居。任二站點 x 與 y 之間有 恰好一條 路徑。

任一站點 x 能夠產生封包 (資料片段) 並能傳送它至其他的站點 y ，站點 y 稱為此封包的 目標。這個封包必須依下述方式，沿著 x 至 y 的唯一路徑 (unique path) 傳送。考慮一持有封包的站點 z ，該封包的目標為 y ($z \neq y$)。在這個狀況中，站點 z 將：

1. 執行一 路由程序 (routing procedure)，此程序找出由 z 至 y 的唯一路徑上 z 的鄰居，並且
2. 將該封包傳送至該鄰居。

然而，各站點的記憶體是有限的，無法儲存 SIB 中所有連結的列表以供路由程序使用。

你的任務是為 SIB 實作一個路由的方法，由兩個程序構成。

- 第一個程序的輸入為 n ，SIB 中所有連結的列表，以及一整數 $k \geq n - 1$ 。此程序賦予每個站點一 0 到 k 的唯一整數，稱為 標記 (label)。
- 第二個程序為路由程序，在所有站點被賦予標記後將被配置到各站點。它 僅 有下列輸入：
 - s ，目前持有一封包的某個站點 標記，
 - t ，該封包的目標站點 標記 ($t \neq s$)，
 - c ，一個列表記錄 s 的所有鄰居 標記。

此程序應回傳該封包將被傳送至的 s 的鄰居 標記。

在某個子任務中，你的解能獲得的分數是依據所有站點標記中的最大值而定 (一般來說，愈小愈好)。

實作細節

你應實作下列程序：

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

- n : SIB 中的站點數量。
- k : 標記可使用的最大值。
- u 與 v : 二大小為 $n - 1$ 的陣列，用來描述連結。對每個 i ($0 \leq i \leq n - 2$)，連結 i 接通編號

$u[i]$ 與 $v[i]$ 的站點。

- 此程序應回傳一大小為 n 的陣列 L 。對每個 i ($0 \leq i \leq n - 1$)， $L[i]$ 為站點 i 被賦予的標記。陣列 L 的每個元素皆為 0 至 k 的唯一整數。

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

- s : 持有封包的某個站點標記。
- t : 該封包的目標站點標記。
- c : 一陣列記錄 s 所有鄰居的標記。陣列 c 依遞增順序排序。
- 此程序應回傳該封包將被傳送至的 s 的鄰居的標記。

每筆測資包含一或多個獨立的情況 (即不同的 SIB 描述)。對每筆測資，令其包含 r 個情況，有一呼叫上列程序的 **program** 將被執行恰好兩次，詳述如下。

在第一次執行 program 時：

- 程序 label 被呼叫 r 次。
- 回傳的所有標記將被評分系統儲存，且
- `find_next_station` 不會被呼叫。

在第二次執行 program 時：

- `find_next_station` 可能被呼叫多次。每次呼叫時會選擇 任意 一個情境，且該情境下程序 label 回傳的所有標記會被做為 `find_next_station` 之輸入。
- label 不會被呼叫。

特別一提，第一次執行 program 時若有任何以 `static` 或 `global` 變數方式儲存的資訊，在程序 `find_next_station` 中皆不能使用。

範例

考慮下列呼叫：

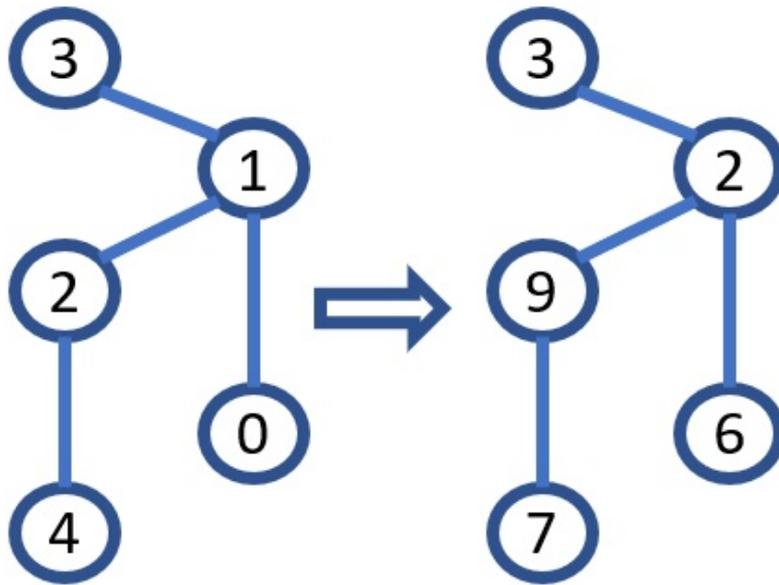
```
label(5, 10, [0, 1, 1, 2], [1, 2, 3, 4])
```

共有 5 個站點以及 4 個連結接通站點對 $(0, 1)$ 、 $(1, 2)$ 、 $(1, 3)$ 以及 $(2, 4)$ 。每個標記可為 0 至 $k = 10$ 的整數。

為了回報下面的標記方式：

Index	Label
0	6
1	2
2	9
3	3
4	7

程序 `label` 應回傳 `[6, 2, 9, 3, 7]`。下圖中的數字為站點編號 (左側) 以及賦予給每個站點的標記 (右側)。



假設標記已被如上圖描述的方式賦予，考慮下列的呼叫：

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

這表示擁有此封包的站點標記為 `9`，而目標站點標記為 `6`。此路徑上所有的站點標記為 `[9, 2, 6]`。因此，此呼叫需回傳 `2`，即此封包下一個被傳送至的站點標記 (即編號 `1` 的站點)。

考慮另一個可能的呼叫：

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

此程序應回傳 `3`，因為目標站點的標記為 `3` 且為標記為 `2` 的站點的鄰居，故應直接收到該封包。

條件限制

- $1 \leq r \leq 10$

對每次 label 的呼叫:

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

對於每次呼叫 find_next_station，其輸入選自之前任意一次所呼叫到的 label，考慮它所產生的標記，然後

- s 與 t 是兩個不同站點的標記。
- c 是標記為 s 的站點的所有鄰居的標記所組成的序列，依遞增順序。

對於每一個測試案例，所有的情況併計後，傳送至 find_next_station 的 c 陣列長度總和不超過 100 000。

子任務

1. (5 points) $k = 1000$ ，每個站點至多 2 個鄰居。
2. (8 points) $k = 1000$ ，連結 i 接通站點 $i + 1$ 與 $\lfloor \frac{i}{2} \rfloor$ 。
3. (16 points) $k = 1\,000\,000$ ，至多一個站點擁有超過 2 個鄰居。
4. (10 points) $n \leq 8, k = 10^9$
5. (61 points) $k = 10^9$

在子任務 5 中你可獲得部分分數。令 m 為在所有情況中 label 回傳的標記最大值。你在此子任務的得分將依下表計算：

Maximum label	Score
$m \geq 10^9$	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

範例評分程式

此範例評分程式依下述格式讀取輸入：

- line 1: r

接下來有 r 個區塊，每個描述單一情況。每個區塊的格式如下：

- line 1: $n \ k$
- line $2 + i$ ($0 \leq i \leq n - 2$): $u[i] \ v[i]$
- line $1 + n$: q : 呼叫 find_next_station 的次數。
- line $2 + n + j$ ($0 \leq j \leq q - 1$): $z[j] \ y[j] \ w[j]$: 第 j 次呼叫 find_next_station 需要的

站點 編號 (indices)：站點 $z[j]$ 持有一封包，站點 $y[j]$ 為該封包之目標，而 $w[j]$ 為 $z[j]$ 至 $y[j]$ 的唯一路徑上 $z[j]$ 的鄰居。

此範例評分程式以下述格式輸出結果：

- line 1: m

接下來依序有 r 個區塊，對應到輸入的情況。每個區塊的格式如下：

- line $1 + j$ ($0 \leq j \leq q - 1$): 站點編號，其 標記 為此情況下第 j 次呼叫 `find_next_station` 的回傳值。

注意，範例評分程式執行時必定會呼叫 `label` 與 `find_next_station`。