



## ステーション (stations)

シンガポールのインターネットバックボーン(Singapore's Internet Backbone : SIB) は  $n$  個のステーションで構成されており, ステーションは  $0$  から  $n - 1$  まで 番号 づけられている.  $n - 1$  個の双方向なリンクがあり,  $0$  から  $n - 2$  まで番号づけられている. それぞれのリンクは  $2$  個の異なるステーションを繋いでいる. リンクによって直接的に繋がれている  $2$  個のステーションを隣接していると呼ぶ.

ステーション  $x$  からステーション  $y$  へのパスとは, 互いに異なるステーションの列  $a_0, a_1, \dots, a_p$  で,  $a_0 = x, a_p = y$  かつパス内の全ての連続する  $2$  個のステーションが隣接しているものである. 任意のステーション  $x$  から他の任意のステーション  $y$  へのパスはただ  $1$  つ 存在する.

任意のステーション  $x$  はパケット(データの断片)を作り, 他の任意のステーション  $y$  に送信することができる. このステーション  $y$  をパケットの **ターゲット** と呼ぶ. パケットは  $x$  から  $y$  への唯一のパスを経由しなければならない. 現在, パケットがステーション  $z$  にあり, ターゲットがステーション  $y$  ( $z \neq y$ ) であるとす. この状態でステーション  $z$  では以下の処理が行われる.

- $z$  から  $y$  への唯一のパス上にあり,  $z$  に隣接しているステーションを決定する **ルーティングプロシージャ** を実行する.
- パケットを隣接するステーションに転送する.

しかし, ステーションが使用できるメモリは限られており, ルーティングプロシージャで用いるためのSIB上の全てのリンクのリストを記憶することはできない.

あなたの仕事はSIBのルーティングシステムを実装することであり, ルーティングシステムは  $2$  個のプロシージャで構成されている.

- $1$  つ目のプロシージャには  $n$  と SIB上のリンクのリスト, 整数  $k \geq n - 1$  が入力として与えられる. このプロシージャはそれぞれステーションに  $0$  以上  $k$  以下の互いに異なる 整数の ラベル 割り当てる.
- $2$  つ目のプロシージャはルーティングプロシージャで, 割り当てられたラベルの情報を利用する. 以下の入力だけ 与えられる.
  - $s$ : 現在パケットを持っているステーションの ラベル.
  - $t$ : パケットのターゲットステーションの ラベル ( $t \neq s$ ).
  - $c$ :  $s$  に隣接している全てのステーションの ラベル の列.

このプロシージャはパケットが転送されるべき  $s$  に隣接するステーションの ラベル を返さなければならない.

小課題において, あなたの解答の点数はステーションに割り当てられたラベルの最大値に依存している(一般的に小さければ小さいほど良い).

## 実装の詳細

あなたは以下のプロシーダを実装しなさい。

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

- $n$ : SIB のステーションの個数.
- $k$ : 使用できるラベルの値の最大値.
- $u, v$ : 長さ  $n - 1$  の配列でリンクを表している. 各  $i$  ( $0 \leq i \leq n - 2$ ) において, リンク  $i$  はステーション  $u[i]$  と  $v[i]$  を繋いでいる.
- このプロシーダは長さ  $n$  の配列  $L$  を返さなければならない. 各  $i$  ( $0 \leq i \leq n - 1$ ) において,  $L[i]$  はステーション  $i$  に割り当てられたラベルである.  $L$  の要素は互いに異なり,  $0$  以上  $k$  以下でなければならない.

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

- $s$ : パケットを持っているステーションのラベル.
- $t$ : パケットのターゲットステーションのラベル.
- $c$ :  $s$  に隣接している全てのステーションのラベルの列を表す配列.  $c$  は昇順に並んでいる.
- このプロシーダはパケットが転送されるべき  $s$  に隣接するステーションのラベルを返さなければならない.

各テストケースは 1 個以上の独立なシナリオ(すなわち, 異なる SIB の状態)を含んでいる.  $r$  個のシナリオを含んでいるテストケースは, 上記のプロシーダを呼び出す プログラム を以下のようにちょうど 2 回実行する.

プログラムの 1 回目の実行は次の通りである.

- `label` プロシーダが  $r$  回呼び出され, 返されたラベルが採点システムによって保存される.
- `find_next_station` は呼び出されない.

プログラムの 2 回目の実行は次の通りである.

- `find_next_station` 複数回呼び出される. 各呼び出しにおいて, 恣意的にある シナリオが選ばれ, そのシナリオにおける `label` プロシーダの呼び出しによって返されたラベルが `find_next_station` の入力として用いられる.
- `label` は呼び出されない.

特に, プログラムの 1 回目の実行で `static` 変数または `global` 変数に保存された任意の情報は `find_next_station` プロシーダ内で使用することはできない.

## 入出力例

次の呼び出しを考える.

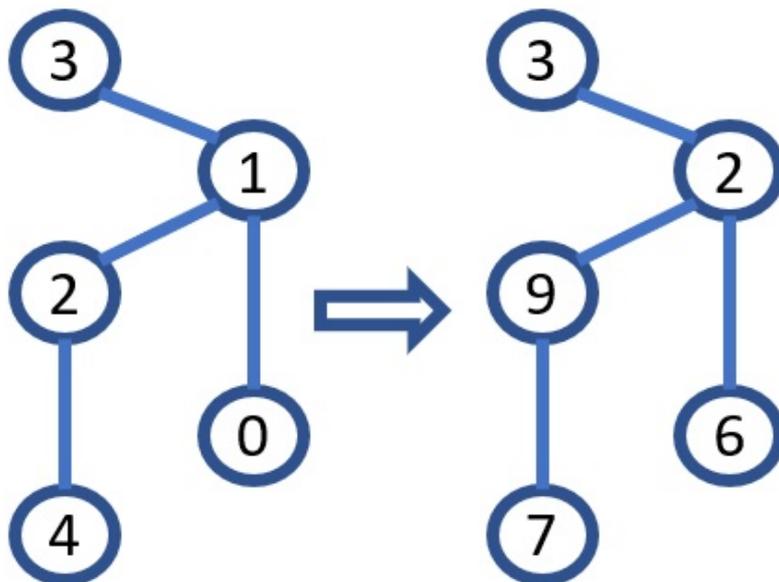
```
label(5, 10, [0, 1, 1, 2], [1, 2, 3, 4])
```

全部で 5 個のステーションがあり, 4 個のリンクはステーションの組  $(0, 1)$ ,  $(1, 2)$ ,  $(1, 3)$ ,  $(2, 4)$  をそれぞれ繋いでいる. 各ラベルは 0 以上の  $k = 10$  以下の整数である.

以下のようなラベルの割り当てを報告するために,

番号	ラベル
0	6
1	2
2	9
3	3
4	7

label プロシージャは  $[6, 2, 9, 3, 7]$  を返さなければならない. 以下の画像の数字は番号(左側)と割り当てられたラベル(右側)を表している.



ラベルが上記の説明のように割り当てられたとして, 次の呼び出しを考える.

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

これはパケットを持っているステーションのラベルは 9 であり, ターゲットステーションのラベルは 6 であることを意味している. ターゲットステーションへのパスのステーションのラベルは  $[9, 2, 6]$  である. したがって, この呼び出しはパケットが転送されるべきステーション(1 と番号づけられている)のラベルである 2 を返すべきである.

別の可能な呼び出しを考える.

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

ラベルが 3 であるターゲットステーションはラベルが 2 であるステーションに隣接しているので、パケットを直接受け取るべきである。したがって、このプロシージャは 3 を返すべきである。

## 制約

- $1 \leq r \leq 10$

各 label の呼び出しに関して

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

find\_next\_station の各呼び出しに関して、それまでの label の呼び出しの中から恣意的に選ばれたデータに基づいて、入力作られる。ラベルが作られたとして、以下のようなになる。

- $s$  と  $t$  は異なる 2 個のステーションのラベルである。
- $c$  は  $s$  に隣接している全てのステーションのラベルの列を表す配列であり、昇順に並んでいる。

各テストケースにおいて、find\_next\_station プロシージャで用いられる配列  $c$  の長さの合計は、全てのシナリオを合わせて 100 000 を超えない。

## 小課題

1. (5 点)  $k = 1000$ , 隣接するステーションが 2 個より多いようなステーションは存在しない。
2. (8 点)  $k = 1000$ , リンク  $i$  はステーション  $i + 1$  とステーション  $\lfloor \frac{i}{2} \rfloor$  を繋いでいる。
3. (16 点)  $k = 1\,000\,000$ , 隣接するステーションが 2 個より多いようなステーションは高々 1 個である。
4. (10 点)  $n \leq 8, k = 10^9$
5. (61 点)  $k = 10^9$

小課題 5 では、あなたは部分点を得ることができる。  $m$  を全てのシナリオにおいて label によって返されたラベルの最大値とする。この小課題におけるあなたの点数は以下の表に応じて計算される。

ラベルの最大値	点数
$m \geq 10^9$	0
$2000 \leq m < 10^9$	$50 \cdot \log_{5 \cdot 10^5} \left( \frac{10^9}{m} \right)$
$1000 < m < 2000$	50
$m \leq 1000$	61

## 採点プログラムのサンプル

採点プログラムのサンプルは以下の形式で入力を読み込む。

- 1 行目:  $r$

$r$  個のブロックが続き, 各ブロックは 1 個のシナリオを表している. 各ブロックの形式は以下のとおりである.

- 1 行目:  $n \ k$
- $2 + i$  行目 ( $0 \leq i \leq n - 2$ ):  $u[i] \ v[i]$
- $1 + n$  行目:  $q$ : `find_next_station` を呼び出す回数.
- $2 + n + j$  行目 ( $0 \leq j \leq q - 1$ ):  $z[j] \ y[j] \ w[j]$ :  $j$  番目の `find_next_station` の呼び出しに関するステーションの番号である. ステーション  $z[j]$  がパケットを持っており, ステーション  $y[j]$  がパケットのターゲットで, ステーション  $w[j]$  がパケットを次に転送すべきステーションである.

採点プログラムのサンプルは以下の形式で答えを出力する.

- 1 行目:  $m$

合計  $r$  個のブロックがシナリオの入力の直後にそれぞれ出力される. 各ブロックの形式は以下のとおりである.

- $1 + j$  行目 ( $0 \leq j \leq q - 1$ ): このシナリオにおける  $j$  番目の `find_next_station` の戻り値であるようなラベルを持つステーションの番号.

採点プログラムのサンプルはそれぞれの実行は `label` と `find_next_station` 両方を呼び出すことに注意せよ.