

ایستگاه‌ها (stations)

زیرساخت اینترنت سنگاپور (SIB) شامل n ایستگاه است که با اندیس‌های 0 تا $n - 1$ شماره گذاری شده‌اند. علاوه‌براین، $n - 1$ لینک بدون جهت با شماره‌های 0 تا $n - 2$ وجود دارد. هر لینک دو ایستگاه متمایز را به هم وصل می‌کند. دو ایستگاه که با لینک به یکدیگر متصل هستند را همسایه می‌نامیم.

یک مسیر از ایستگاه x به ایستگاه y دنباله‌ای از ایستگاه‌های a_0, a_1, \dots, a_p است که $a_0 = x, a_p = y$ و هر دو ایستگاه مجاوری در این مسیر، همسایه هستند. بین هر دو ایستگاهی مانند x و y دقیقاً یک مسیر وجود دارد.

هر ایستگاه مانند x می‌تواند یک بسته (قطعه‌ای از داده) ایجاد کند و به هر ایستگاه دیگری مانند y که مقصد بسته نامیده می‌شود، بفرستد. این بسته باید از مسیر یکتایی که از x به y وجود دارد حرکت کند. ایستگاه z را در نظر بگیرید که در حال حاضر بسته‌ای با مقصد y دارد. در این وضعیت، ایستگاه z باید:

1. یک رویه مسیریابی را اجرا کند که این رویه، همسایه‌ای از z که در مسیر یکتا از z به y هست را مشخص کند.
2. بسته را به این همسایه بفرستد.

البته ایستگاه‌ها حافظه محدودی دارند و نمی‌توانند تمام لینک‌های SIB را برای رویه مسیریابی نگه‌داری کنند.

وظیفه شما این است که یک برنامه مسیریابی برای SIB طراحی کنید که شامل دو تابع زیر باشد.

- تابع اول، n ، لیست لینک‌های SIB و عدد طبیعی $k \geq n - 1$ را به عنوان ورودی می‌گیرد. این تابع به هر ایستگاه یک لیبل صحیح و یکتا بین 0 تا k (شامل این دو عدد) نسبت می‌دهد.
- تابع دوم رویه مسیریابی را پیاده‌سازی می‌کند. این تابع بر روی تمام ایستگاه‌ها، بعد از این‌که لیبل‌هایشان مشخص شود، مستقر می‌شود. به این تابع تنها ورودی‌های زیر داده می‌شود:

- s ، لیبل ایستگاه فعلی که بسته را نگه داشته است.
- t ، لیبل ایستگاه مقصد بسته ($t \neq s$).
- c ، لیستی از لیبل‌های همسایه‌های s .

این تابع باید لیبل همسایه‌ای از s که بسته باید به آن فرستاده شود را برگرداند.

در یک زیرمسئله، نمره شما به مقدار بزرگ‌ترین لیبلی که به هر ایستگاهی نسبت داده‌اید، بستگی دارد (در مجموع، هر چه این مقدار کمتر باشد بهتر است).

جزئیات پیاده‌سازی

شما باید تابع زیر را پیاده‌سازی کنید:

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

- n : تعداد ایستگاه‌ها در SIB.
- k : بزرگ‌ترین لیبلی که می‌توان استفاده کرد.
- u و v : آرایه‌هایی به طول $n - 1$ که لینک‌ها را مشخص می‌کنند. برای هر i ($0 \leq i \leq n - 2$)، لینک i ایستگاه‌هایی با اندیس $u[i]$ و $v[i]$ را به هم متصل می‌کند.
- این تابع باید آرایه L به طول n را برگرداند. برای هر i ($0 \leq i \leq n - 1$)، $L[i]$ لیبلی است که به ایستگاه i نام نسبت داده شده است. تمام اعضای آرایه L باید بین 0 تا k (شامل این دو عدد) و یکتا باشند.

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

- s : لیبیل ایستگاهی که بسته را نگه داشته است.
- t : لیبیل ایستگاه مقصد بسته.
- c : آرایه‌ای که لیست لیبیل‌های تمام همسایه‌های s است. آرایه c به صورت صعودی مرتب شده است.
- این تابع باید لیبیل همسایه‌ای از s که بسته باید به آن فرستاده شود را برگرداند.

هر تست کیس شامل یک یا تعدادی سناریو مستقل می‌شود (در واقع شامل تعدادی ساختار متفاوت برای SIB است). برای هر تست کیس شامل r سناریو، یک برنامه که تابع بالا را صدا می‌زند دقیقاً دوبار به صورت زیر اجرا می‌شود.

طی اجرای اول برنامه:

- تابع $label$ ، r مرتبه صدا زده می‌شود.
- لیبیل‌های برگردانده شده توسط سیستم ارزیاب ذخیره می‌شوند. و
- تابع $find_next_station$ صدا زده نمی‌شود.

طی اجرای دوم برنامه:

- تابع $find_next_station$ ممکن است چندین مرتبه صدا زده شود. در هر مرتبه، یک سناریو **دلخواه** انتخاب می‌شود و لیبیل‌هایی که در تابع $label$ برگردانده شده‌اند، به عنوان ورودی در $find_next_station$ استفاده می‌شوند.
- تابع $label$ صدا زده نمی‌شود.

در واقع اطلاعاتی که در متغیرهای $global$ و $static$ در اجرای اول ذخیره شده‌اند، در تابع $find_next_station$ در دسترس نمی‌باشند.

مثال

فراخوانی زیر را در نظر بگیرید:

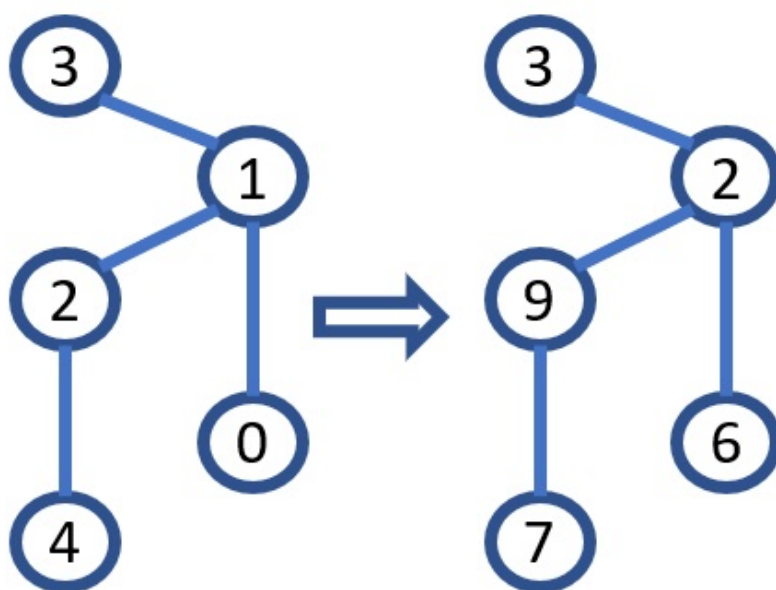
```
label(5, 10, [0, 1, 1, 2], [1, 2, 3, 4])
```

در مجموع 5 ایستگاه وجود دارد که 4 لینک اندیس‌های $(0, 1)$ ، $(1, 2)$ ، $(1, 3)$ و $(2, 4)$ را به هم متصل کرده است. هر لیبیل می‌تواند عدد صحیحی بین 0 تا $k = 10$ باشد.

برای اینکه لیبیل گذاری زیر را برگردانیم:

لیبل	اندیس
6	0
2	1
9	2
3	3
7	4

تابع label باید [6, 2, 9, 3, 7] را برگرداند. در تصویر زیر اندیس‌ها (در تصویر چپ) و لیبل‌های نسبت‌داده شده (در تصویر راست) قابل مشاهده است.



فرض کنید که لیبل‌ها به صورتی که در بالا گفته شده، نسبت داده شده‌اند و فراخوانی زیر صورت می‌گیرد:

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

بدین معنا است که ایستگاهی که بسته را نگه داشته است لیبل 9 و ایستگاه مقصد نیز لیبل 6 دارد. لیبل ایستگاه‌هایی که در مسیر به ایستگاه مقصد هستند، [9, 2, 6] است. از این رو، فراخوانی باید 2 برگرداند که لیبل ایستگاهی است که بسته باید به آن فرستاده شود (که اندیس 1 دارد).

فراخوانی زیر را در نظر بگیرید:

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

این تابع باید 3 برگرداند. زیرا مقصد لیبل 3 دارد که همسایه ایستگاه با لیبل 2 است و باید بسته را به صورت مستقیم دریافت کند.

محدودیت‌ها

- $1 \leq r \leq 10$

برای هر فراخوانی label:

- $2 \leq n \leq 1000$

- $k \geq n - 1$

- $0 \leq u[i], v[i] \leq n - 1$ (برای هر $0 \leq i \leq n - 2$)

برای هر فراخوانی تابع find_next_station، ورودی از یک فراخوانی دلخواهی که قبلاً روی label داشته‌ایم، می‌آید. لیبل‌هایی که در این سناریو تولید شده است را در نظر بگیرید. آنگاه:

- s و t لیبل‌های دو ایستگاه متفاوت هستند.

- c دنباله‌ای از لیبل‌های همسایه‌های ایستگاه s به ترتیب صعودی است.

برای تمام تست‌کیس‌های مجموع کل سناریوها، مجموع طول آرایه c که به تابع find_next_station داده می‌شود از 100 000 عبور نمی‌کند.

زیرمسئله‌ها

1. (5 امتیاز) $k = 1000$ ، هیچ ایستگاهی بیش از 2 همسایه ندارد.
2. (8 امتیاز) $k = 1000$ ، لینک i ایستگاه $i + 1$ و $\lfloor \frac{i}{2} \rfloor$ را به هم متصل می‌کند.
3. (16 امتیاز) $k = 1\,000\,000$ ، حداکثر یک ایستگاه بیش‌تر از 2 همسایه دارد.
4. (10 امتیاز) $n \leq 8$ ، $k = 10^9$
5. (61 امتیاز) $k = 10^9$

در زیرمسئله ۵ شما می‌توانید بخشی از نمره (partial score) را بگیرید. فرض کنید m بزرگ‌ترین لیبلی باشد که توسط تابع label برای تمام سناریوها برگردانده شده است. نمره شما برای این زیرمسئله بر اساس جدول زیر محاسبه می‌شود:

بزرگ‌ترین لیبل	نمره
$m \geq 10^9$	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

ارزیاب نمونه

ارزیاب نمونه ورودی را در قالب زیر می‌خواند

- خط 1: r

در ادامه r بلاک می‌آیند که هر کدام یک سناریو را توضیح می‌دهند. فرمت هر بلاک به صورت زیر است:

- خط 1: n k
 - خط $2 + i$: $(0 \leq i \leq n - 2)$ $u[i]$ $v[i]$
 - خط $1 + n$: q : تعداد دفعاتی که تابع `find_next_station` صدا زده می‌شود.
 - خط $2 + n + j$: $(0 \leq j \leq q - 1)$ $z[j]$ $y[j]$ $w[j]$: **اندیس** ایستگاه‌هایی که در j امین فراخوانی تابع `find_next_station` استفاده می‌شوند. ایستگاه $z[j]$ بسته را نگه داشته است، ایستگاه $y[j]$ مقصد بسته است، و ایستگاه $w[j]$ ایستگاهی است که بسته باید به آن در این مرحله فرستاده شود.
- ارزیاب نمونه خروجی را در قالب زیر چاپ می‌کند:
- خط 1: m
- n بلاک که هر کدام به ترتیب مربوط به یک سناریو در ورودی است. فرمت هر بلاک به صورت زیر است:
- خط $1 + j$: $(0 \leq j \leq q - 1)$ **اندیس** ایستگاهی که **لیبل** آن توسط j امین فراخوانی تابع `find_next_station` در این سناریو برگردانده شده است.
- توجه کنید که هر اجرای ارزیاب نمونه هر دو تابع `label` و `find_next_station` را فرا می‌خواند.