Csomópontok (stations)

Szingapúr Internetes Gerinchálózata (SzIG) n csomópontból áll, melyek 0-tól $n-1$-ig vannak **sorszámozva**. Van $n-1$ kétirányú kapcsolat is, 0-tól $n-2$-ig számozva. Minden kapcsolat két különböző csomópontot köt össze. Két közvetlenül kapcsolódó csomópontot szomszédosnak nevezünk.

Egy x csomópontból y-ba vezető útvonalon különböző csomópontok olyan $a_0, a_1, \cdots, a_p$ sorozatát értjük, melyre $a_0 = x$, $a_p = y$, és bármely két az útvonalon egymást követő csomópont szomszédos. Bármely x csomópontból bármely másik y csomópontba **pontosan egy** útvonal van.

Bármely x csomópont készíthet egy adatcsomagot és elküldheti bármely másik y csomópontnak, amit a csomag **célpontjának** hívunk. A csomagot végig kell irányítani az x-ből y-ba vezető egyetlen útvonalon az alábbiak szerint. Tekintsünk egy z csomópontot ahol éppen van egy csomag, melynek célpontja y ($z \neq y$). Az állomás

1. lefuttat egy **útkereső függvényt**, amely meghatározza z-nek azt a szomszédját, amely a z-ből y-ba vezető egyetlen útvonalon van, és
2. továbbítja a csomagot ennek a szomszédnak.

Azonban a csomópontok memóriája korlátozott, nem tárolják a SzIG összes kapcsolatát az útkereső függvény futtatásához.

Téged kértek meg, hogy készítsd el az útkereső rendszert a SzIG-hez, amely két függvényből áll.

- Az első függvény bemenetként megkapja az n-et, a SzIG összes kapcsolatának listáját és egy $k \geq n-1$ egész számot. Minden csomóponthoz hozzá kell rendelnie egy **egyedi** c **címkét** (egész számot), $0 \leq c \leq k$.

- A második az útkereső függvény, amit a csomópontok futtatni fognak a címkék kiosztása után. **Csak** a következő bemeneteket kapja:

  - s, a csomópont **címkéje** ahol éppen egy csomag van,
  - t, a csomag célpontjának **címkéje** ($t \neq s$),
  - c, egy lista, ami s szomszédainak **címkéit** tartalmazza.

  Visszatérési értéke legyen s azon szomszédjának **címkéje**, ahova a csomagot továbbítani kell.

A megoldásod pontszáma függ a maximális címke értéktől, amit kiosztottál a csomópontok között (minél kisebb, annál jobb).

Megvalósítás

A következő függvényeket kell elkészítened:

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

- n: a SzIG csomópontjainak száma.
- k: a legnagyobb címke érték amit használhatsz.
- u és v: $n-1$ méretű tömbök amik a kapcsolatokat írják le. Minden i-re ($0 \leq i \leq n-2$), az i. kapcsolat az $u[i]$ és a $v[i]$ sorszámú csomópontokat köti össze.
- A függvény visszatérési értéke egy n méretű L tömb legyen! Minden i-re ($0 \leq i \leq n-1$) az i sorszámú csomópont az $L[i]$ címkét kapja. L elemei legyenek különböző, 0 és k közti egész számok!

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

- s: a csomópont címkéje, ahol épp egy csomag van.
- t: a csomag célpontjának címkéje.
- c: egy tömb, ami s szomszédainak címkéit tartalmazza. A c tömb növekvően rendezett.
- A függvény visszatérési értéke s azon szomszédjának címkéje legyen, ahova a csomagot továbbítani kell!

Minden teszteset egy vagy több független tesztet tartalmaz (különböző SzIG leírásokat). Egy tesztesetre amely r tesztet tartalmaz, a fenti függvényeket meghívó **program** pontosan kétszer lesz lefuttatva, az alábbiak szerint.

A program első futtatása során:

- a `label` függvény r-szer lesz meghívva,
- a visszaadott címkéket az értékelő rendszer elmenti, és
- a `find_next_station` nem lesz meghívva.

A program második futtatása során:

- a `find_next_station` függvény többször is meg lehet hívva. Minden híváskor kiválasztásra kerül egy tetszőleges teszt, és a `label` erre a tesztre adott címkéi alpján kapja a bemenetét a `find_next_station`.
- a `label` nem lesz meghívva.

Figyelem, bármi információ, amit statikus vagy globális vátozókba mentesz a program első futása során, nem lesz elérhető a `find_next_station` függvényben.

Példa

Tekintsük a következő hívást:

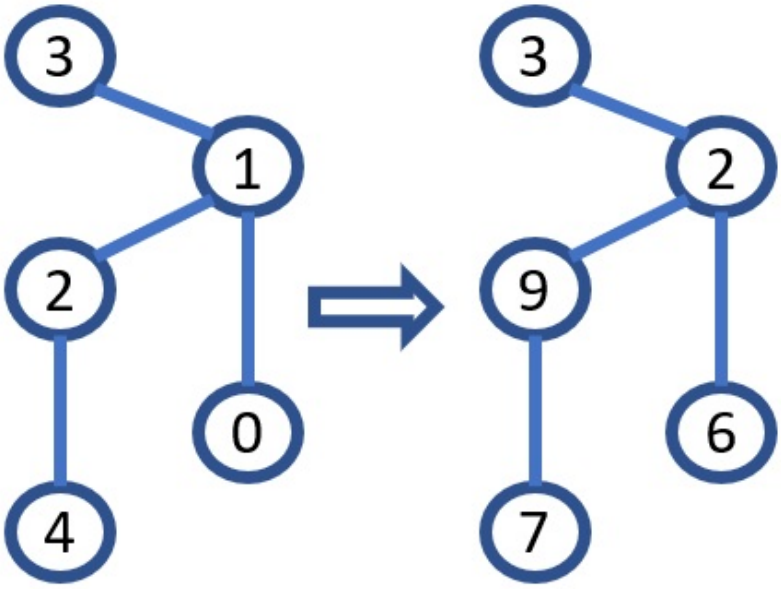
```
label(5, 10, [0, 1, 1, 2], [1, 2, 3, 4])
```

Összesen 5 csomópont van, és 4 kapcsolat köti össze a $(0, 1)$, $(1, 2)$, $(1, 3)$ és $(2, 4)$ csomópontpárokat. Minden címke 0 és $k = 10$ közti egész szám lehet.

Az alábbi címkézés közléséhez:

Sorszám	Címke
0	6
1	2
2	9
3	3
4	7

a `label` függvénynek a $[6, 2, 9, 3, 7]$ tömböt kell visszaadnia. Az alábbi ábra bal oldala mutatja a sorszámokat, a jobb oldal pedig az egyes csomópontokhoz rendelt címkéket.



Tegyük fel, hogy a címkéket a fentiek szerint osztottuk ki, és tekintsük a következő hívást:

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

Ez azt jelenti, hogy a csomag épp a 9-es címkéjű csomópontnál van, a célpont címkéje pedig 6. A célponthoz vezető úton a csomópontok címkéi $[9, 2, 6]$. Így a függvénynek 2-t kell visszaadnia, ami annak a csomópontnak a címkéje, ahova a csomagot továbbítani kell (a sorszáma 1).

Tekintsünk egy másik lehetséges hívást:

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

A függvénynek 3-at kell visszaadnia, mert a 3-as címkéjű célpont szomszédja a 2-es címkéjű csomópontnak, így közvetlenül megkaphatja a csomagot.

Korlátok

- $1 \leq r \leq 10$

A `label` minden hívására:

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

A `find_next_station` minden hívására a bemenet egy tetszőlegesen választott korábbi `label` hívásból származik. Tekintsük a címkéket, amiket kiosztott.

- s és t két különböző csomópont címkéje.
- az s címkéjű csomópont szomszédai címkéinek sorozata c, növekvő sorrendben.

Minden tesztesetben a `find_next_station` függvénynek átadott c tömbök összhossza legfeljebb $100\ 000$ az összes tesztre együttvéve. <!--* $1 \leq d \leq n - 1$

- $0 \leq s, t \leq k$
- $s \neq t$
- $0 \leq c[i] \leq k$ ($0 \leq i \leq d - 1$)
- $c[i - 1] < c[i]$ ($1 \leq i \leq d - 1$)
- A `find_next_station`-nek átadott c tömbök összhossza legfeljebb $100\ 000$ az összes tesztet együttvéve.-->

Részfeladatok

1. (5 pont) $k = 1000$, egy csomópontnak sincs több, mint 2 szomszédja.
2. (8 pont) $k = 1000$, az i-edik kapcsolat az $i + 1$-edik és a $\left\lfloor \frac{i}{2} \right\rfloor$-edik csomópontot köti össze.
3. (16 pont) $k = 1\ 000\ 000$, legfeljebb egy csomópontnak van több, mint 2 szomszédja.
4. (10 pont) $n \leq 8$, $k = 10^9$
5. (61 pont) $k = 10^9$ <!-- 1. (5 points) $k = 1000$, no station has more than 2 neighbours.
6. (8 points) $k = 1000$, link i connects stations $i + 1$ and $\left\lfloor \frac{i}{2} \right\rfloor$.
7. (16 points) $k = 1\ 000\ 000$, at most one station has more than 2 neighbours.
8. (10 points) $n \leq 8$, $k = 10^9$
9. (61 points) $k = 10^9$ -->

Az 5. részfeladatban részpontszámokat is kaphatsz. Legyen m a legnagyobb címke érték amit a `label` visszaadott, az összes tesztet tekintve. A részfeladatra kapott pontszámodat a a következő

táblázat határozza meg:

Maximális címke	Pontszám
$m \geq 10^9$	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

Minta értékelő

A minta értékelő az alábbi formában olvassa a bemenetet:

- Az 1. sor: r

A következő r blokk írja le a teszteket. A blokkok formátuma a következő:

- Az 1. sor: $n \ k$
- A $2+i$. sor ($0 \leq i \leq n-2$): $u[i] \ v[i]$
- Az $1+n$. sor: a `find_next_station` hívásainak száma, q.
- A $2+n+j$. sor ($0 \leq j \leq q-1$): $z[j] \ y[j] \ w[j]$: a `find_next_station` j. hívásában érintett csomópontok **sorszáma**: a $z[j]$ csomópontban van a csomag, az $y[j]$ csomópont a csomag célpontja, és $w[j]$ az a csomópont, ahova továbbítani kell a csomagot.

A minta értékelő a következő formában írja ki a választ:

- Az 1. sor: m

A következő r blokk a bemenetben megadott teszteknek felel meg. A blokkok formátuma a következő:

- Az $1+j$. sor ($0 \leq j \leq q-1$): Annak a csúcsnak a **sorszáma**, amelynek a **címkéjét** a `find_next_station` (ezen a teszten belüli) j-edik hívása visszaadta.

Figyelem, a minta értékelő egy futtatása meghívja a `label` és a `find_next_station` függvényt is.

Csomópontok (stations)

Szingapúr Internetes Gerinchálózata (SzIG) n csomópontból áll, melyek 0-tól $n-1$-ig vannak **sorszámozva**. Van $n-1$ kétirányú kapcsolat is, 0-tól $n-2$-ig számozva. Minden kapcsolat két különböző csomópontot köt össze. Két közvetlenül kapcsolódó csomópontot szomszédosnak nevezünk.

Egy x csomópontból y-ba vezető útvonalon különböző csomópontok olyan $a_0, a_1, \cdots, a_p$ sorozatát

értjük, melyre $a_0 = x$, $a_p = y$, és bármely két az útvonalon egymást követő csomópont szomszédos. Bármely x csomópontból bármely másik y csomópontba **pontosan egy** útvonal van.

Bármely x csomópont készíthet egy adatcsomagot és elküldheti bármely másik y csomópontnak, amit a csomag **célpontjának** hívunk. A csomagot végig kell irányítani az x-ből y-ba vezető egyetlen útvonalon az alábbiak szerint. Tekintsünk egy z csomópontot ahol éppen van egy csomag, melynek célpontja y ($z \neq y$). Az állomás

1. lefuttat egy **útkereső függvényt**, amely meghatározza z-nek azt a szomszédját, amely a z-ből y-ba vezető egyetlen útvonalon van, és
2. továbbítja a csomagot ennek a szomszédnak.

Azonban a csomópontok memóriája korlátozott, nem tárolják a SzIG összes kapcsolatát az útkereső függvény futtatásához.

Téged kértek meg, hogy készítsd el az útkereső rendszert a SzIG-hez, amely két függvényből áll.

- Az első függvény bemenetként megkapja az n-et, a SzIG összes kapcsolatának listáját és egy $k \geq n - 1$ egész számot. Minden csomóponthoz hozzá kell rendelnie egy c **címkét** (egyedi egész számot), $0 \leq c \leq k$.

- A második az útkereső függvény, amit a csomópontok futtatni fognak a címkék kiosztása után. **Csak** a következő bemeneteket kapja:

    - s, a csomópont **címkéje** ahol éppen egy csomag van,
    - t, a csomag célpontjának **címkéje** ($t \neq s$),
    - c, egy lista, ami s szomszédainak **címkéit** tartalmazza.

    Visszatérési értéke legyen s azon szomszédjának **címkéje**, ahova a csomagot továbbítani kell.

A megoldásod pontszáma függ a maximális címke értéktől, amit kiosztottál a csomópontok között (minél kisebb, annál jobb).

Megvalósítás

A következő függvényeket kell elkészítened:

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

- n: a SzIG csomópontjainak száma.
- k: a legnagyobb címke érték amit használhatsz.
- u és v: $n - 1$ méretű tömbök amik a kapcsolatokat írják le. Minden i-re ($0 \leq i \leq n - 2$), az i. kapcsolat az $u[i]$ és a $v[i]$ sorszámú csomópontokat köti össze.
- A függvény visszatérési értéke egy n méretű L tömb legyen! Minden i-re ($0 \leq i \leq n - 1$) az i sorszámú csomópont az $L[i]$ címkét kapja. L elemei legyenek különböző, 0 és k közti egész számok!

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

- s: a csomópont címkéje, ahol épp egy csomag van.
- t: a csomag célpontjának címkéje.
- c: egy tömb, ami s szomszédainak címkéit tartalmazza. A c tömb növekvően rendezett.
- A függvény visszatérési értéke s azon szomszédjának címkéje legyen, ahova a csomagot továbbítani kell!

Minden teszteset egy vagy több független tesztet tartalmaz (különböző SzIG leírásokat). Egy tesztesetre amely r tesztet tartalmaz, a fenti függvényeket meghívó **program** pontosan kétszer lesz lefuttatva, az alábbiak szerint.

A program első futtatása során:

- a `label` függvény r-szer lesz meghívva,
- a visszaadott címkéket az értékelő rendszer elmenti, és
- a `find_next_station` nem lesz meghívva.

A program második futtatása során:

- a `find_next_station` függvény többször is meg lehet hívva. Minden híváskor kiválasztásra kerül egy tetszőleges teszt, és a `label` erre a tesztre adott címkéi alpján kapja a bemenetét a `find_next_station`.
- a `label` nem lesz meghívva.

Figyelem, bármi információ, amit statikus vagy globális vátozókba mentesz a program első futása során, nem lesz elérhető a `find_next_station` függvényben.

Példa

Tekintsük a következő hívást:

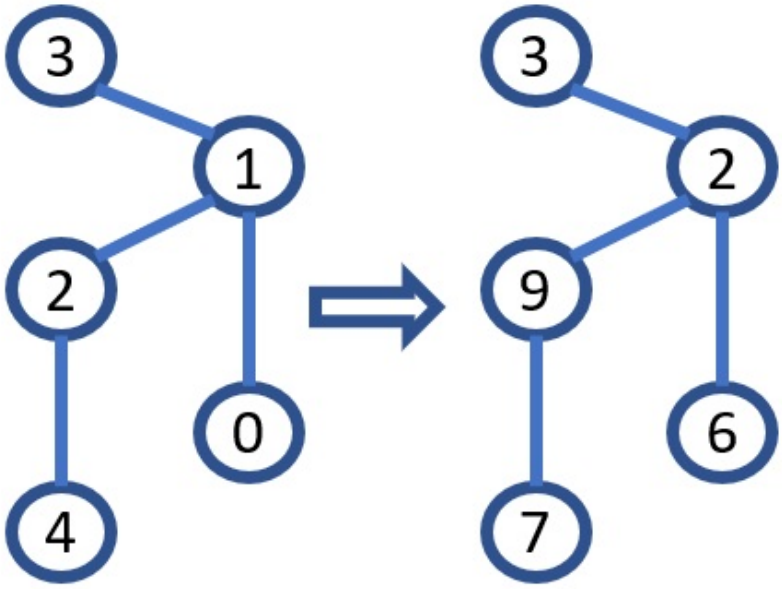
```
label(5, 10, [0, 1, 1, 2], [1, 2, 3, 4])
```

Összesen 5 csomópont van, és 4 kapcsolat köti össze a $(0, 1)$, $(1, 2)$, $(1, 3)$ és $(2, 4)$ csomópontpárokat. Minden címke 0 és $k = 10$ közti egész szám lehet.

Az alábbi címkézés közléséhez:

Sorszám	Címke
0	6
1	2
2	9
3	3
4	7

a `label` függvénynek a $[6, 2, 9, 3, 7]$ tömböt kell visszaadnia. Az alábbi ábra bal oldala mutatja a sorszámokat, a jobb oldal pedig az egyes csomópontokhoz rendelt címkéket.



Tegyük fel, hogy a címkéket a fentiek szerint osztottuk ki, és tekintsük a következő hívást:

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

Ez azt jelenti, hogy a csomag épp a 9-es címkéjű csomópontnál van, a célpont címkéje pedig 6. A célponthoz vezető úton a csomópontok címkéi $[9, 2, 6]$. Így a függvénynek 2-t kell visszaadnia, ami annak a csomópontnak a címkéje, ahova a csomagot továbbítani kell (a sorszáma 1).

Tekintsünk egy másik lehetséges hívást:

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

A függvénynek 3-at kell visszaadnia, mert a 3-as címkéjű célpont szomszédja a 2-es címkéjű csomópontnak, így közvetlenül megkaphatja a csomagot.

Korlátok

- $1 \leq r \leq 10$

A `label` minden hívására:

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

A `find_next_station` minden hívására, ha a c tömb hossza d:

- $1 \leq d \leq n - 1$
- $0 \leq s, t \leq k$
- $s \neq t$
- $0 \leq c[i] \leq k$ ($0 \leq i \leq d - 1$)
- $c[i - 1] < c[i]$ ($1 \leq i \leq d - 1$)
- A `find_next_station`-nek átadott c tömbök összhossza legfeljebb $100\,000$ az összes tesztet együttvéve.

Részfeladatok

1. (5 pont) $k = 1000$, egy csomópontnak sincs több, mint 2 szomszédja.
2. (8 pont) $k = 1000$, az i-edik kapcsolat az $i + 1$-edik és a $\left\lfloor \frac{i}{2} \right\rfloor$-edik csomópontot köti össze.
3. (16 pont) $k = 1\,000\,000$, legfeljebb egy csomópontnak van több, mint 2 szomszédja.
4. (10 pont) $n \leq 8$, $k = 10^9$
5. (61 pont) $k = 10^9$ <!-- 1. (5 points) $k = 1000$, no station has more than 2 neighbours.
6. (8 points) $k = 1000$, link i connects stations $i + 1$ and $\left\lfloor \frac{i}{2} \right\rfloor$.
7. (16 points) $k = 1\,000\,000$, at most one station has more than 2 neighbours.
8. (10 points) $n \leq 8$, $k = 10^9$
9. (61 points) $k = 10^9$ -->

Az 5. részfeladatban részpontszámokat is kaphatsz. Legyen m a legnagyobb címke érték amit a `label` visszaadott, az összes tesztet tekintve. A részfeladatra kapott pontszámodat a a következő táblázat határozza meg:

Maximális címke	Pontszám
$m \geq 10^9$	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

Minta értékelő

A minta értékelő az alábbi formában olvassa a bemenetet:

- Az $1.$ sor: r

A következő r blokk írja le a teszteket. A blokkok formátuma a következő:

- Az $1.$ sor: $n \ \ k$
- A $2 + i.$ sor ($0 \leq i \leq n - 2$): $u[i] \ \ v[i]$
- Az $1 + n.$ sor: a `find_next_station` hívásainak száma, q.
- A $2 + n + j.$ sor ($0 \leq j \leq q - 1$): $z[j] \ \ y[j] \ \ w[j]$: a `find_next_station` $j.$ hívásában érintett csomópontok **sorszámai**: a $z[j]$ csomópontban van a csomag, az $y[j]$ csomópont a csomag célpontja, és $w[j]$ az a csomópont, ami $z[j]$ után következik a $z[j]$-ből $y[j]$-be vezető egyetlen útvonalon.

A minta értékelő a következő formában írja ki a választ:

- Az $1.$ sor: m

A következő r blokk a bemenetben megadott teszteknek felel meg. A blokkok formátuma a következő:

- Az $1 + j.$ sor ($0 \leq j \leq q - 1$): Annak a csúcsnak a **sorszáma**, amelynek a **címkéjét** a `find_next_station` (ezen a teszten belüli) j-edik hívása visszaadta.

Figyelem, a minta értékelő egy futtatása meghívja a `label` és a `find_next_station` függvényt is.