

Stations (stations)

Singapurško Internetno Hrbtenico (SIH) tvori n točk, katerim so dodeljene **oznake** od 0 do $n - 1$. Obstaja še $n - 1$ dvosmernih povezav, označenih z 0 do $n - 2$. Vsaka povezava povezuje dve ločeni točki. Dve točki, kateri povezuje ena povezava, sta sosednji.

Pot od točke x do točke y je zaporedje točk a_0, a_1, \dots, a_p , tako da $a_0 = x$, $a_p = y$, in vsaki dve zaporedni točki na poti sta sosednji. Obstaja **natanko ena** pot od katere koli točke x do katere koli druge točke y .

Vsaka točka x lahko ustvari paket (košček podatka) in ga pošlje kateri koli drugi točki y , kar imenujemo paketov **cilj**. Ta paket se mora usmeriti po edinstveni poti od x do y : Oglejmo si točko z , ki trenutno drži paket, katerega ciljna točka je y ($z \neq y$). Ta točka

1. izvede **usmerjevalno proceduro**, ki določi soseda točke z , ki leži na edinstveni poti od z do y , in
2. usmeri paket k temu sosеду.

Vendar: točke imajo omejeno količino pomnilnika in ne hranijo celotnega seznama povezav v SIH za izvedbo usmerjevalne procedure.

Implementiramo usmerjevalno shemo za SIH, ki se sestoji iz dveh funkcij.

- Prva funkcija sprejme n , seznam povezav v SIH, in celo število $k \geq n - 1$. Vsaki točki dodeli edinstveno celoštevilsko **oznako** med 0 in k , vključujoče.
- Druga funkcija je usmerjevalna procedura, ki se jo razpošlje na vse točke po tem, ko se jim dodeli oznake. Dobi **zgolj** naslednje vhode:
 - s , **oznaka** točki, ki trenutno zadržuje paket,
 - t , **oznaka** ciljne točke ($t \neq s$),
 - c , seznam **oznak** vseh sosedov s .

Rezultat naj bo **oznaka** soseda s , kateremu naj se paket pošlje naprej.

Dodatno, rezultat tvoje rešitve je odvisen od največje oznake dodeljene katerikoli točki (v splošnem, manjše je boljše).

Podrobnosti implementacije

Implementiraj naslednji funkciji:

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

- n : število točk SIH.
- k : največja oznaka, ki se jo lahko uporabi.
- u in v : polji velikosti $n - 1$, ki opisujeta povezave. Za vsak i ($0 \leq i \leq n - 2$), povezava i povezuje točki z oznakama $u[i]$ in $v[i]$.
- Ta funkcija vrne eno polje L velikosti n . Za vsak i ($0 \leq i \leq n - 1$) je $L[i]$ oznaka dodeljena točki z indeksom i . Vsi elementi polja L so med seboj različni in med 0 in k , vključujoče.

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

- s : oznaka točke, kjer se nahaja paket.
- t : oznaka ciljne točke.
- c : polje oznak vseh sosedov točke s . Polje c je urejeno v naraščajočem vrstnem redu.
- Funkcija vrne oznako točke s , h kateri naj se paket pošlje.

Vsak testni primer vključuje enega ali več neodvisnih scenarijev (t.j. različnih opisov SIH). Za testni primer, ki vsebuje r scenarijev, je **program**, ki kliče zgornje funkcije, klican natanko dvakrat:

Med prvim klicem programa:

- funkcija `label` je poklicana r -krat,
- vrnjene oznake si shrani ocenjevalni sistem, in
- `find_next_station` ni poklicana.

Med drugim klicem programa:

- `find_next_station` se lahko večkrat pokliče,
- oznake, uporabljene pri klicih `find_next_station` so oznake, katere je vrnil klic `label` za **poljubno izbran** scenarij iz prvega klica, in
- `label` ni poklicana.

Pomembno: nobena od informacij, shranjenih v statične ali globalne spremenljivke v prvem klicu programa, ni dostopna znotraj funkcije `find_next_station`.

Primeri

Predpostavimo naslednji klic.

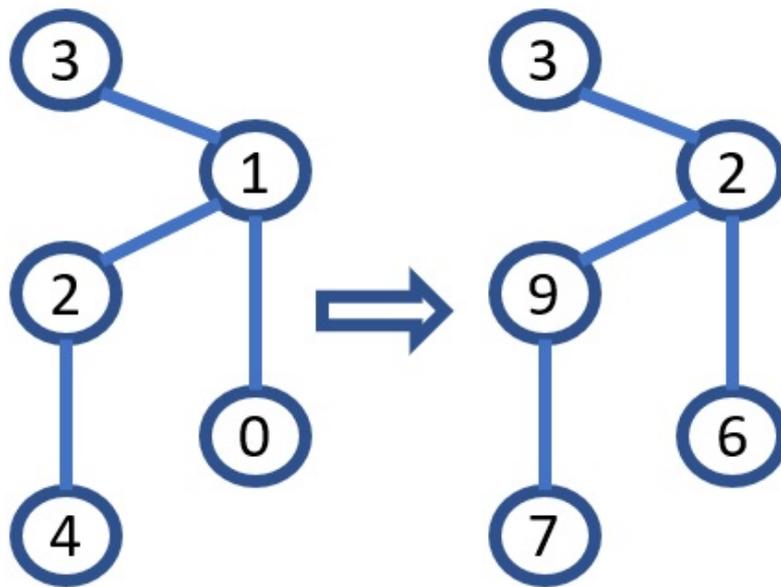
```
label(5, 10, [0, 1, 1, 2], [1, 2, 3, 4])
```

Obstaja 5 točk in 4 povezav, ki povezujejo točke z indeksi (0,1), (1,2), (1,3) in (2,4). Vsaka oznaka je lahko med 0 in $k = 10$.

Da zavedemo naslednjo označitev:

| Indeks | oznaka |
|--------|--------|
| 0 | 6 |
| 1 | 2 |
| 2 | 9 |
| 3 | 3 |
| 4 | 7 |

funkcija `label` vrne `[6, 2, 9, 3, 7]`. Na sliki so na levi prikazani indeksi, na desni pa točkam dodeljene oznake.



Predpostavi, da so oznake dodeljene kot je zgoraj opisano in obravnavajmo naslednji klic:

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

To pomeni, da ima točka, ki trenutno zadržuje paket, oznako 9, in da ima ciljna točka oznako 6. Oznake točk na poti do cilja so `[9, 2, 6]`. Zatorej bi klic naj vrnil 2, saj je to oznaka točke, h kateri je potrebno usmeriti paket (ima indeks 1).

Obravnavajmo drug možen klic:

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

Funkcija vrne 3, saj ima ciljna točka oznako 3, ki je sosednja točki z oznako 2, in zatorej paket prejme direktno.

Omejitve

- $1 \leq r \leq 10$

Za vsak klic `label`:

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

Za vsak klic `find_next_station`:

- s in t sta oznaki dveh različnih točk.
- c je seznam oznak vseh sosedov točke z oznako s , v naraščajočem vrstnem redu.
- Oznake pripadajo enemu od predhodnih klicev funkcije `label`.

Skupna dolžina vseh seznamov c , podanih funkciji `find_next_station`, v nobenem od testnih primerov ne preseže 100 000.

Podnaloge

1. (5 točk) $k = 1000$, nobena točka nima več kot 2 soseda.
2. (8 točk) $k = 1000$, povezava i povezuje točki $i + 1$ in $\lfloor \frac{i}{2} \rfloor$.
3. (16 točk) $k = 1\,000\,000$, največ ena točka ima čez kot 2 soseda.
4. (10 točk) $n \leq 8$, $k = 10^9$
5. (61 točk) $k = 10^9$

Pri 5. podnalogi je možen delni rezultat. Naj bo m največja oznaka, ki jo je vrnil klic `label` v vseh scenarijih. Število točk za to podnalogo bo izračunano po sledeči tabeli:

| Največja oznaka | Točke |
|----------------------|--|
| $m \geq 10^9$ | 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 |

Vzorčni ocenjevalnik

Vzorčni ocenjevalnik bere vhod v naslednjem formatu:

- vrstica 1: r

Sledi r opisov posameznih scenarijev v naslednjem formatu:

- vrstica 1: $n \ k$
- vrstica $2 + i$ ($0 \leq i \leq n - 2$): $u[i] \ v[i]$

- vrstica $1 + n$: q : število klicev `find_next_station`.
- line $2 + n + j$ ($0 \leq j \leq q - 1$): $z[j]$ $y[j]$ $w[j]$: **indeksi** točk, vpletenih v i -ti klic `find_next_station`. Paket je na točki $z[j]$, točka $y[j]$ je ciljna točka, $w[j]$ pa je točka, na katero naj se usmeri paket. The station $z[j]$ holds the packet, the station $y[j]$ is the packet's target, and the station $w[j]$ is the station that the packet should be forwarded to.

Vzorčni ocenjevalnik izpiše tvoje odgovore v naslednjem formatu:

- vrstica 1: m

Sledi r opisov posameznih scenarijev v naslednjem formatu:

- vrstica $1 + j$ ($0 \leq j \leq q - 1$): **indeks** točke, katere **oznako** je vrnil j -ti klic `find_next_station` v tem scenariju.

Upoštevaj, da vsak zagon vzorčnega ocenjevalnika kliče `label` in `find_next_station`.