Sfinksas mīkla

Diženā Sfinksa uzdod jums mīklu. Jums tiek dots grafs ar N virsotnēm, kas ir numurētas no 0 līdz $N-1$. Grafā ir M šķautnes, kas sanumurētas no 0 līdz $M-1$. Katra šķautne savieno atšķirīgu virsotņu pāri un tā ir neorientēta. Konkrēti, katram j no 0 līdz $M-1$ (ieskaitot) šķautne j savieno virsotnes $X[j]$ un $Y[j]$. Katru virsotņu pāri savieno ne vairāk kā viena šķautne. Divas virsotnes sauc par **kaimiņu** virsotnēm, ja tās savieno šķautne.

Virsotņu virkne $v_0, v_1, \ldots, v_k$ (visiem $k \geq 0$) tiek saukta par **ceļu**, ja katras divas secīgas virsotnes v_l un v_{l+1} (visiem l, kur $0 \leq l < k$) ir kaimiņu virsotnes. Mēs sakām, ka ceļš $v_0, v_1, \ldots, v_k$ **savieno** virsotnes v_0 un v_k. Jums dotajā grafā katru virsotņu pāri savieno kāds ceļš.

Ir $N+1$ krāsas, kas numurētas no 0 līdz N. Krāsa N ir īpaša, to sauc par **Sfinksas krāsu**. Katra virsotne ir kādā krāsā. Konkrēti, virsotne i ($0 \leq i < N$) ir krāsā $C[i]$. Vairākas virsotnes var būt vienā krāsā, un var būt krāsas, kādā nav neviena virsotne. Neviena virsotne nav Sfinksas krāsā, tas ir, $0 \leq C[i] < N$ ($0 \leq i < N$).

Ceļu $v_0, v_1, \ldots, v_k$ (visiem $k \geq 0$) sauc par **monohromatisku**, ja visas tā virsotnes ir vienādā krāsā, tas ir, $C[v_l] = C[v_{l+1}]$ (visiem l, kur $0 \leq l < k$). Turklāt mēs sakām, ka virsotnes p un q ($0 \leq p < N$, $0 \leq q < N$) ir tajā pašā **monohromatiskajā komponentē** tad un tikai tad, ja tās savieno monohromatisks ceļš.

Jūs zināt virsotnes un šķautnes, bet jūs nezināt, kāda krāsā ir katra virsotne. Jūs vēlaties uzzināt virsotņu krāsas, veicot **pārkrāsošanas eksperimentus**.

Pārkrāsošanas eksperimentā jūs varat pārkrāsot patvaļīgi daudz virsotņu. Konkrēti, lai veiktu pārkrāsošanas eksperimentu, jūs vispirms izvēlaties masīvu E garumā N, kur visiem i ($0 \leq i < N$), $E[i]$ ir no -1 līdz N **ieskaitot**. Tad i-tās virsotnes krāsa kļūst $S[i]$, kur $S[i]$ vērtība ir:

- $C[i]$, tas ir, i-tās virsotnes sākotnējā krāsa, ja $E[i] = -1$, vai
- $E[i]$, pretējā gadījumā.

Ņemiet vērā, ka tas nozīmē, ka pārkrāsošanā varat izmantot Sfinksas krāsu.

Visbeidzot, pēc visu virsotņu i krāsas nomainīšanas uz $S[i]$ ($0 \leq i < N$), varenā Sfinksa paziņo monohromatisko komponenšu skaitu grafā. Jaunais krāsojums tiek izmantots tikai šim konkrētajam pārkrāsošanas eksperimentam, tādēļ, **pēc eksperimenta beigām, visu virsotņu krāsas atgriežas sākuma stāvoklī**.

Jūsu uzdevums ir noteikt grafa virsotņu krāsas, veicot ne vairāk kā $2\,750$ pārkrāsošanas eksperimentus. Varat arī saņemt daļēju punktu skaitu, ja katram kaimiņu virsotņu pārim pareizi nosakāt, vai tā virsotnes ir vienādā krāsā.

Implementēšanas detaļas

Jums jāimplementē šāda procedūra:

```
std::vector<int> find_colours(int N,
    std::vector<int> X, std::vector<int> Y)
```

- N: virsotņu skaits grafā.
- X, Y: masīvi garumā M, kas apraksta šķautnes.
- Šai procedūrai jāatgriež masīvs G garumā N, kas reprezentē virsotņu krāsas grafā.
- Šo procedūru izsauks tieši vienu reizi katrā testā.

Iepriekš minētā procedūra var izsaukt šādu procedūru, lai veiktu pārkrāsošanas eksperimentus:

```
int perform_experiment(std::vector<int> E)
```

- E: ir masīvs garumā N, kas definē, kā virsotnes ir jāpārkrāso.
- Šī procedūra atgriež monohromatisko komponenšu skaitu pēc virsotņu pārkrāsošanas atbilstoši E.
- Šo procedūru var izsaukt ne vairāk kā $2\,750$ reizes.

Vērtētājprogramma **nav adaptīva**, tas ir, virsotņu krāsas tiek fiksētas, pirms tiek izsaukta procedūra `find_colours` .

Ierobežojumi

- $2 \leq N \leq 250$
- $N - 1 \leq M \leq \frac{N \cdot (N-1)}{2}$
- $0 \leq X[j] < Y[j] < N$ visiem j, kur $0 \leq j < M$.
- $X[j] \neq X[k]$ vai $Y[j] \neq Y[k]$ visiem j un k, kur $0 \leq j < k < M$.
- Visus virsotņu pārus savieno kāds ceļš.
- $0 \leq C[i] < N$ visiem i, kur $0 \leq i < N$.

Apakšuzdevumi

Apakšuzdevums	Punkti	Papildu ierobežojumi
1	3	$N = 2$
2	7	$N \leq 50$
3	33	Grafs ir ceļš: $M = N - 1$ un virsotnes j un $j + 1$ ir kaimiņu virsotnes ($0 \leq j < M$).
4	21	Grafs ir pilns: $M = \frac{N \cdot (N-1)}{2}$ un jebkuras divas virsotnes ir kaimiņu virsotnes.
5	36	Bez papildu ierobežojumiem.

Katrā apakšuzdevumā varat iegūt daļēju punktu skaitu, ja jūsu programma katram kaimiņu virsotņu pārim pareizi nosaka, vai tā virsotnes ir vienādā krāsā.

Precīzāk, jūs iegūstat pilnu apakšuzdevuma punktu skaitu, ja visos testos masīvs G, ko atgriež `find_colours`, ir tieši tāds pats kā masīvs C (t.i. $G[i] = C[i]$ visiem i, kur $0 \leq i < N$). Citādi, jūs iegūstat 50% no apakšuzdevuma punktu skaita, ja sekojošie nosacījumi ir spēkā visiem apakšuzdevuma testiem:

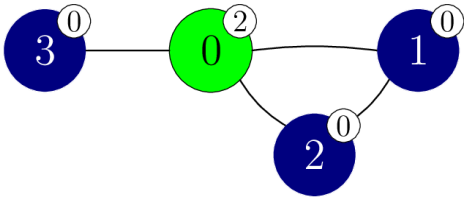
- $0 \leq G[i] < N$ visiem i, kur $0 \leq i < N$;
- Visiem j, kur $0 \leq j < M$:
  - $G[X[j]] = G[Y[j]]$ tad un tikai tad, ja $C[X[j]] = C[Y[j]]$.

Piemērs

Aplūkosim šādu procedūras izsaukumu:

```
find_colours(4, [0, 1, 0, 0], [1, 2, 2, 3])
```

Šajā piemērā pieņemsim, ka virsotņu (slēptās) krāsas ir $C = [2, 0, 0, 0]$. Šis scenārijs ir parādīts nākamajā attēlā. Krāsas ir papildus attēlotas ar ciparіem uz baltām etiķetēm, kas pievienotas katrai virsotnei.



Procedūra var izsaukt `perform_experiment` šādi:

```
perform_experiment([-1, -1, -1, -1])
```

Šajā izsaukumā neviena virsotne netiek pārkrāsota, jo visas virsotnes saglabā savas sākotnējās krāsas.

Aplūkosim virsotni 1 un virsotni 2. Tās abas ir krāsā 0, un ceļš $1, 2$ ir monohromatisks. Rezultātā virsotnes 1 un 2 atrodas vienā monohromatiskajā komponentē.

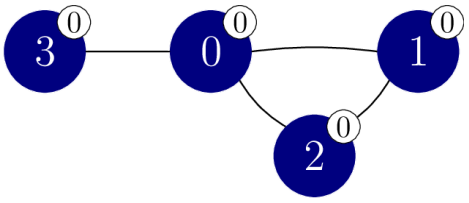
Aplūkosim virsotni 1 un virsotni 3. Lai gan abas ir krāsā 0, tās ir dažādās monohromatiskās komponentēs, jo nav monohromatiska ceļa, kas tās savieno.

Kopumā ir 3 monohromatiskās komponentes ar virsotnēm $\{0\}$, $\{1, 2\}$, un $\{3\}$. Tādēļ šis izsaukums atgriež vērtību 3.

Tagad procedūra var izsaukt `perform_experiment` šādi.

```
perform_experiment([0, -1, -1, -1])
```

Šajā izsaukumā tikai virsotne 0 ir pārkrāsota krāsā 0, kā rezultātā tiek iegūts krāsojums, kas parādīts nākamajā attēlā.

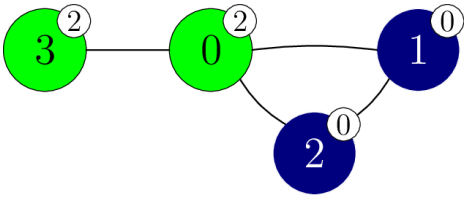


Šis izsaukums atgriež 1, jo visas virsotnes pieder vienai un tai pašai monohromatiskajai komponentei. Tagad varam izsecināt, ka virsotnes 1, 2 un 3 ir krāsā 0.

Pēc tam procedūra var izsaukt `perform_experiment` sekojoši.

```
perform_experiment([-1, -1, -1, 2])
```

Šajā izsaukumā virsotne 3 ir pārkrāsota krāsā 2, kā rezultātā tiek iegūts krāsojums, kas parādīts nākamajā attēlā.



Šis izsaukums atgriež 2, jo ir 2 monohromatiskās komponentes ar virsotnēm $\{0, 3\}$ un $\{1, 2\}$ attiecīgi. Mēs varam secināt, ka virsotne 0 ir krāsā 2.

Pēc tam procedūra `find_colours` atgriež masīvu $[2, 0, 0, 0]$. Tā kā $C = [2, 0, 0, 0]$, tiek piešķirts pilns punktu skaits.

Ņemiet vērā, ka ir arī vairākas atgriežamas vērtības, par kurām tiktu piešķirti 50% no punktu skaita, piemēram $[1, 2, 2, 2]$ vai $[1, 2, 2, 3]$.

Paraugvērtētājs

Ievaddatu formāts:

```
N  M
C[0]  C[1] ... C[N-1]
X[0]  Y[0]
X[1]  Y[1]
...
X[M-1]  Y[M-1]
```

Izvaddatu formāts:

```
L  Q
G[0]  G[1] ... G[L-1]
```

L ir masīva G, ko atgriež `find_colours`, garums un Q ir `perform_experiment` izsaukumu skaits.