

Užduotis: Navigacija

 3 sek.  512 MB

Duotas **jungus bekryptis paprastas kaktuso grafas**¹ su $N \leq 1000$ viršūnių ir M briaunų. Jo viršūnės yra spalvotos (spalvos pažymėtos neneigiamais sveikaisiais skaičiais nuo 0 iki 1499). Iš pradžių visos viršūnės turi spalvą 0. **Deterministinis robotas be atminties**² tyrinėja grafą judėdamas tarp viršūnių. Jis turi aplankyti visas viršūnes bent kartą, ir tada išsijungti.

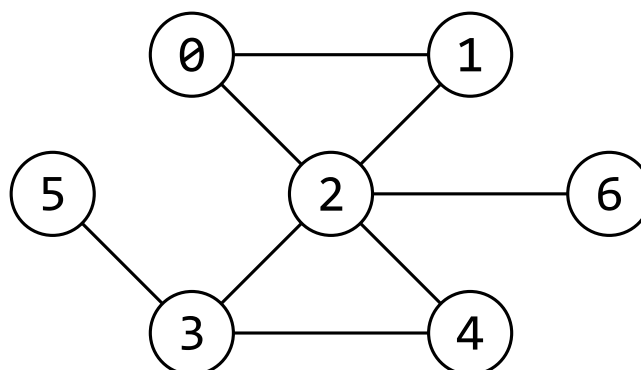
Robotas pradeda judėti iš tam tikros viršūnės, kuri gali būti bet kuri iš grafo viršūnių. Kiekvieno žingsnio metu robotas mato dabartinės viršūnės spalvą, ir visų kaimyninių viršūnių spalvas **tam tikra tvarka, fiksuota dabartinei viršūnei** (t.y., aplankydamas tą pačią viršūnę dar kartą robotas matys tą pačią viršūnių tvarką, net jei jų spalvos pasikeitė). Robotas atlieka vieną iš šių dviejų veiksmų:

1. Nusprendžia išsijungti.
2. Parenka naują (ar galimai tą pačią) spalvą dabartinei viršūnei, ir pasirenka kaimyninę viršūnę, į kurią jis planuoja judėti. Kaimyninė viršūnė yra nusakoma indeksu nuo 0 iki $D - 1$, kur D yra kaimyninių viršūnių skaičius.

Antruoju atveju, dabartinė viršūnė yra perdažoma (arba galimai lieka tos pačios spalvos) ir robotas pajuda į pasirinktą kaimyninę viršūnę. Tai kartojasi tol, kol robotas išsijungia, arba pasiekia iteracijų limitą. Jei robotas neaplanko visų viršūnių iki išsijungimo, arba pasiekia iteracijų limitą, jis pralaimi. Iteracijų limitas lygus $L = 3000$ žingsnių.

Sukurkite roboto strategiją, kuri galėtų išspręsti šią užduotį bet kuriam tokiam kaktuso grafui. Be to, pabandykite minimizuoti skirtingų spalvų skaičių, kurį naudoja jūsų sprendimas. Spalva 0 įtraukta į naudojamas spalvas.

¹Jungus bekryptis paprastas kaktuso grafas yra jungus bekryptis paprastas grafas (kiekviena viršūnė yra pasiekama iš bet kurios kitos viršūnės; briaunos neturi krypties; nėra kilpų, o kiekvieną viršūnių porą jungia ne daugiau nei viena briauna), kuriame kiekviena briauna priklauso daugiausiai vienam paprastajam ciklui (paprastasis ciklas yra ciklas, kuriame kiekviena viršūnė pasikartoja daugiausiai vieną kartą). Žemiau esanti iliustracija yra pavyzdys.





²Robotas yra deterministinis ir be atminties, jei jo veiksmai priklauso tik nuo dabartinių duomenų (t.y., jis nesaugo jokios informacijos iš praėjusių žingsnių), ir su tais pačiais pradiniais duomenimis jis visada pasirenka tą patį veiksmą.



Realizacija

Roboto strategija realizuojama šia funkcija:

```
std::pair<int, int> navigate(int currColor, std::vector<int> adjColors)
```

Ji gauna du parametrus: dabartinės viršūnės spalvą bei visų kaimyninių viršūnių spalvas (tam tikra tvarka). Ji turi grąžinti porą, kurios pirmasis elementas yra nauja dabartinės viršūnės spalva, o antrasis elementas yra kaimyninės viršūnės indeksas, į kurią robotas turėtų judėti. Jei robotas turėtų sustoti ir išsijungti, funkcija turi grąžinti porą $(-1, -1)$.

Ši funkcija bus kviečiama pakartotinai tam, kad būtų parinkti roboto veiksmai. Kadangi ji yra deterministinė, jei `navigate` jau buvo iškviesta su tam tikrais parametrais, ji niekada nebus vėl iškviečiama su tais pačiais parametrais; vietoj to, jo ankstesnė išvesta reikšmė bus pernaudota. Be to, kiekvienas testas gali turėti $T \leq 5$ dalinių testų (su skirtingais grafais ar/ir pradinėmis pozicijomis), kurie gali būti kviečiami tuo pačiu metu (t.y., jūsų programa gali būti iškviesta iš pakaitomis besikeičiančių dalinių testų). Galiausiai, `navigate` gali būti iškviesta **skirtingose** jūsų programos **kopijose** (bet gali būti kviečiama ir toje pačioje programos kopijoje). Bendras jūsų programos kopijų skaičius yra $P = 100$. Dėl šios priežasties, jūsų programa neturi perduoti informacijos tarp skirtingų iškviетimų.



Ribojimai

- $3 \leq N \leq 1000$
- $0 \leq \text{Spalva} < 1500$
- $L = 3000$
- $T \leq 5$
- $P = 100$



Vertinimas

Taškų, kuriuos gausite, už konkrečią dalinę užduotį, dalis S priklausys nuo C – maksimalaus skirtingo spalvų skaičiaus, kurį naudoja jūsų sprendimas (įskaičiuojant spalvą 0) iš visų tos dalinės užduoties dalinių testų ar bet kurios kitos reikalingos dalinės užduoties:

- Jei jūsų sprendimas neįveikia nei vieno dalinio testo, tuomet $S = 0$.
- Jei $C \leq 4$, tuomet $S = 1.0$.



- Jei $4 < C \leq 8$, tuomet $S = 1.0 - 0.6 \frac{C-4}{4}$.
- Jei $8 < C \leq 21$, tuomet $S = 0.4 \frac{8}{C}$.
- Jei $C > 21$, tuomet $S = 0.15$.

Dalinės užduotys

Dalinė užduotis	Taškai	Reikalingos dalinės užduotys	N	Papildomi ribojimai
0	0	—	≤ 300	Pavyzdys.
1	6	—	≤ 300	Grafas yra ciklas. ¹
2	7	—	≤ 300	Grafas yra žvaigždė. ²
3	9	—	≤ 300	Grafas yra takas. ³
4	16	2 – 3	≤ 300	Grafas yra medis. ⁴
5	27	—	≤ 300	Visos viršūnės turi ne daugiau nei 3 kaimynines viršūnes ir viršūnė, kurioje robotas pradeda judėti, turi 1 kaimyninę viršūnę.
6	28	0 – 5	≤ 300	—
7	7	0 – 6	—	—

¹Grafas ciklas turi šias briaunas: $(i, (i + 1) \bmod N)$ visiems $0 \leq i < N$.

²Grafas žvaigždė turi šias briaunas: $(0, i)$ visiems $1 \leq i < N$.

³Grafas takas turi šias briaunas: $(i, i + 1)$ for $0 \leq i < N - 1$.

⁴Medis yra grafas be ciklų.

Pavyzdys

Nagrinėkime paprastąjį grafą iš šios sąlygos iliustracijos, kuris turi $N = 7$, $M = 8$ bei viršūnes $(0, 1)$, $(1, 2)$, $(2, 0)$, $(2, 3)$, $(3, 4)$, $(4, 2)$, $(3, 5)$ ir $(2, 6)$. Papildomai, kadangi elementų tvarka viršūnių kaimynysčių sąrašė yra svarbi, pateikiame šią tvarką lentelėje:



Viršūnė	Kaimyninės viršūnės
0	2, 1
1	2, 0
2	0, 3, 4, 6, 1
3	4, 5, 2
4	2, 3
5	3
6	2

Tarkime, kad robotas pradeda judėti viršūnėje 5. Tai yra viena iš galimų (nesėkmingų) interakcijos sekų:

#	Spalvos	Viršūnė	Kvietinys į navigate	Grąžinta reikšmė
1	0, 0, 0, 0, 0, 0, 0	5	navigate(0, {0})	{1, 0}
2	0, 0, 0, 0, 0, 1, 0	3	navigate(0, {0, 1, 0})	{4, 2}
3	0, 0, 0, 4, 0, 1, 0	2	navigate(0, {0, 4, 0, 0, 0})	{0, 3}
4	0, 0, 0, 4, 0, 1, 0	6	¹ navigate(0, {0})	{1, 0}
5	0, 0, 0, 4, 0, 1, 1	2	navigate(0, {0, 4, 0, 1, 0})	{8, 0}
6	0, 0, 8, 4, 0, 1, 1	0	navigate(0, {8, 0})	{3, 0}
7	3, 0, 8, 4, 0, 1, 1	2	navigate(8, {3, 4, 0, 1, 0})	{2, 2}
8	3, 0, 2, 4, 0, 1, 1	4	navigate(0, {2, 4})	{1, 1}
9	3, 0, 2, 4, 1, 1, 1	3	navigate(4, {1, 1, 2})	{-1, -1}

Šiame pavyzdyje robotas iš viso panaudojo 6 skirtingas spalvas: 0, 1, 2, 3, 4 ir 8 (atkreipkite dėmesį, kad 0 būtų įskaičiuota kaip panaudota spalva, net jeigu robotas nebūtų grąžinęs spalvos 0, nes visų viršūnių pradinė spalva yra 0). Robotas įvykdė 9 iteracijas prieš išsijungdamas. Tačiau, jis neįveikė šio dalinio testo, nes jis išsijungė neaplankęs viršūnės 1.

¹Atkreipkite dėmesį, kad `navigate` 4-os iteracijos metu iš tikrųjų nebūtų iškviesta. Tai yra todėl, kadangi jis yra ekvivalentus iškvietimui 1-oje iteracijoje, tad vertinimo programa paprasčiausiai pakartotinai panaudotų jūsų funkcijos to iškvietimo grąžintą reikšmę. Vis dėlto, ši iteracija vis tiek yra įtraukiama į roboto atliktų interakcijų skaičių.



Pavyzdinė vertinimo programa

Pavyzdinė vertinimo programa nepaleidžia daugybės programos kopijų, todėl visi išskvietimai į navigate bus atliekami toje pačioje jūsų programos kopijoje.

Pradinių duomenų formatas: Pirmiausia T (dalinių testų skaičius) yra perskaitomas. Tada, kiekvienam daliniui testui:

- 1-oji eilutė: du sveikieji skaičiai – N ir M ;
- $2 + i$ -oji eilutė (kiekvienam $0 \leq i < M$): du sveikieji skaičiai – A_i ir B_i , kurie apibūdina dvi viršūnes, kurias jungia briauna i ($0 \leq A_i, B_i < N$).

Pavyzdinė vertinimo programa tada išves skirtingų spalvų skaičių, kurį panaudojo jūsų sprendimas, ir atliktą iteracijų skaičių iki išsijungimo. Kitu atveju, ji išves klaidos pranešimą, jei jūsų sprendimas buvo nesėkmingas.

Pagal nutylėjimą, pavyzdinė vertinimo programa spausdina išsamią informaciją apie tai, ką robotas mato bei daro kiekvienos iteracijos metu. Jūs galite tai išjungti, pakeisdami DEBUG vertę iš `true` į `false`.