

Zadatak Navigacija

 3 sek.  512 MB

Dat je **povezani neusmjereni prosti kaktus graf**¹ sa $N \leq 1000$ čvorova i M ivica. Čvorovi su obojeni (boje su označene sa cijelim ne-negativnim brojevima od 0 do 1499). Na početku, svi čvorovi imaju boju 0. **Deterministički bezmemorijski robot**² istražuje graf krećući se iz jednog čvora u drugi. Mora posjetiti sve čvorove bar jednom i potom se ugasi.

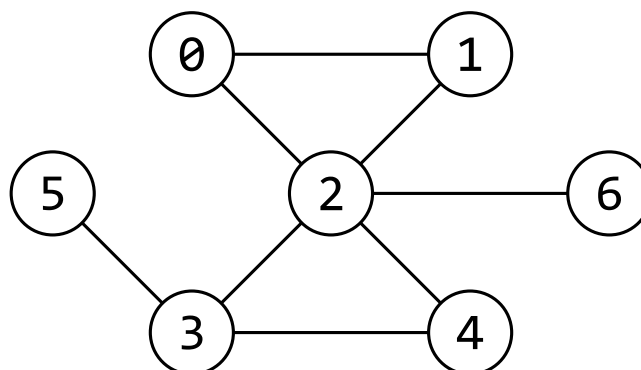
Robot počinje od nekog čvora, koji može biti bilo koji čvor u grafu. Prilikom svakog koraka, robot vidi boju trenutnog čvora i boje svih susjednih čvorova **u nekom fiksnom redoslijedu za trenutni čvor** (odnosno, ponovno posjećivanje čvora će dati robotu isti niz susjednih čvorova, čak i ako su njihove boje drugačije od onih koje su bile prije). Robot može uraditi jednu od dvije sljedeće akcije:

1. Donosi odluku da se ugasi.
2. Bira novu (može i istu) boju za trenutni čvor i bira susjedni čvor na koji će se pomjeriti. Susjedni čvor je jednoznačno označen indeksom od 0 do $D-1$, gdje je D broj susjednih čvorova.

Ako izabere drugu akciju, trenutni čvor se oboji u drugu boju (ili se zadrži postojeća boja) i robot se pomjeri na odabrani susjedni čvor. To se ponavlja sve dok se robot ili ne ugasi ili dok ne dostigne ograničenje na broj iteracija. Robot će pobijediti ako posjeti sve čvorove i onda se ugasi unutar ograničenog broja iteracija od $L = 3000$ koraka (u suprotnom, robot gubi).

Trebate dizajnirati takvu strategiju za robota koja će riješiti problem za bilo kakav kaktus graf. Dodatno, trebate minimizirati broj različitih boja koje vaše rješenje koristi. Boja sa oznakom 0 se uvijek računa kao da je korištena.

¹*Povezani neusmjereni prosti kaktus graf je povezani neusmjereni prosti graf (do svakog čvora se može doći od bilo kojeg drugog čvora; ivice su dvosmjerne; nema povratne petlje na čvorištima ili višestruke ivice) u kojem svaka ivica pripada samo jednom jednostavnom ciklusu (jednostavan ciklus je ciklus koji sadrži svaki čvor samo jednom). Primjer se nalazi na slici ispod.*



²Robot se smatra determinističkim i bezmemorijskim, ako njegova akcija ovisi isključivo o trenutnim ulazima (odnosno, ne čuva informacije između dva koraka), i uvijek bira istu akciju kada su mu dati isti ulazi.

Pojednosti implementacije

Strategija za robota treba biti implementirana u vidu naredne funkcije:

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

Funkcija kao ulaze prima boju trenutnog čvora i boje svih susjednih čvorova (u određenom redoslijedu). Funkcija mora vratiti par, čiji je prvi element nova boja trenutnog čvora, a drugi element je indeks susjednog čvora na koji se robot treba pomjeriti. Ukoliko se želi narediti da se robot ugasi, umjesto da se pomjeri, funkcija bi trebala vratiti par $(-1, -1)$.

Pozivi prema funkciji će se ponavljati da bi se odabrala akcija robota. Obzirom da je robot determinističan, ako je funkcija `navigate` prethodno pozvana sa nekim parametrima, više se neće pozivati sa tim parametrima; umjesto toga, koristiće se vrijednost koju je funkcija prethodno vratila. Dodatno, svaki test može sadržavati $T \leq 5$ podtestova (različiti grafovi i/ili različite startne pozicije) koji se mogu pokretati konkurentno (odnosno, vaš program može dobiti naizmjenične pozive koji se odnose na različite podtestove). Konačno, pozivi ka funkciji `navigate` mogu se desiti u **odvojenim izvršenjima** vašeg programa (ali je moguće da se ponakad dešavaju unutar istog izvršenja). Ukupan broj izvršenja vašeg programa je $P = 100$. Radi svega navedenog, vaš program ne bi trebao prosljeđivati podatke između različitih poziva.

Ograničenja

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

Ocjenjivanje

Dio S poena koje ćete dobiti za podzadatak ovisi o C - maksimalnom broj različitih boja koje vaše rješenje koristi (uključujući boju 0) na bilo kojem testu unutar tog podzadatka ili bilo kojeg drugog zahtjevanog podzadatka:

- Ako vaše rješenje padne na bilo kojem podtestu, onda je $S = 0$.
- Ako je $C \leq 4$, tada je $S = 1.0$.



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



Podzadaci

Podzadatak	Poeni	Zahtjevani podzadatak	N	Dodatna ograničenja
0	0	—	≤ 300	Primjer.
1	6	—	≤ 300	Graf je cikličan. ¹
2	7	—	≤ 300	Graf je zvijezda. ²
3	9	—	≤ 300	Graf je putanja. ³
4	16	2 – 3	≤ 300	Graf je stablo. ⁴
5	27	—	≤ 300	Svi čvorovi imaju najviše 3 susjedna čvora i čvor na kojem robot počinje ima 1 susjedni čvor.
6	28	0 – 5	≤ 300	—
7	7	0 – 6	—	—

¹Ciklični graf ima ivice: $(i, (i + 1) \bmod N)$ for $0 \leq i < N$.

²Zvijezda graf ima ivice: $(0, i)$ for $1 \leq i < N$.

³Putanja graf ima ivice: $(i, i + 1)$ for $0 \leq i < N - 1$.

⁴Stablo je graf bez ciklusa.



Primjer

Posmatrajmo primjer grafa sa slike u postavci zadatka, koji ima $N = 7$, $M = 8$ i ivice $(0, 1)$, $(1, 2)$, $(2, 0)$, $(2, 3)$, $(3, 4)$, $(4, 2)$, $(3, 5)$ i $(2, 6)$. Dodatno, obzirom da su redoslijedi elemenata u listama susjednih čvorova važni, pobrojaćemo ih u narednoj tabeli:



Čvor	Susjedni čvorovi
0	2, 1
1	2, 0
2	0, 3, 4, 6, 1
3	4, 5, 2
4	2, 3
5	3
6	2

Pretpostavimo da robot počinje na čvoru 5. Tada imamo jednu moguću (neuspješnu) sekvencu interakcija:

#	Boje	Čvor	Poziv navigate	Povratna vrijednost
1	0, 0, 0, 0, 0, 0, 0	5	<code>navigate(0, {0})</code>	<code>{1, 0}</code>
2	0, 0, 0, 0, 0, 1, 0	3	<code>navigate(0, {0, 1, 0})</code>	<code>{4, 2}</code>
3	0, 0, 0, 4, 0, 1, 0	2	<code>navigate(0, {0, 4, 0, 0, 0})</code>	<code>{0, 3}</code>
4	0, 0, 0, 4, 0, 1, 0	6	¹ <code>navigate(0, {0})</code>	<code>{1, 0}</code>
5	0, 0, 0, 4, 0, 1, 1	2	<code>navigate(0, {0, 4, 0, 1, 0})</code>	<code>{8, 0}</code>
6	0, 0, 8, 4, 0, 1, 1	0	<code>navigate(0, {8, 0})</code>	<code>{3, 0}</code>
7	3, 0, 8, 4, 0, 1, 1	2	<code>navigate(8, {3, 4, 0, 1, 0})</code>	<code>{2, 2}</code>
8	3, 0, 2, 4, 0, 1, 1	4	<code>navigate(0, {2, 4})</code>	<code>{1, 1}</code>
9	3, 0, 2, 4, 1, 1, 1	3	<code>navigate(4, {1, 1, 2})</code>	<code>{-1, -1}</code>

U ovom slučaju robot je koristio ukupno 6 različitih boja: 0, 1, 2, 3, 4 i 8 (obratite pažnju da se boja 0 broji kao korištena čak i ako robot nikada nije vratio boju 0, obzirom da svi čvorovi počinju sa bojom 0). Robot je izvršio 9 iteracija prije nego što se ugasio. Međutim, bio je neuspješan, budući da se ugasio bez da je posjetio čvor 1.

¹Obratite pažnju da se poziv ka `navigate` u iteraciji 4 ustvari neće desiti. Razlog je to što je poziv ekvivalentan pozivu u iteraciji 1, tako da će ocjenjivač jednostavno iskoristiti vrijednost koju je funkcija vratila u tom pozivu. Međutim, ovo se i dalje računa kao jedna iteracija robota.



Primjer ocjenjivača

Primjer ocjenjivača neće izvršavati vaš program više puta, tako da će svi pozivi ka `navigate` biti unutar jednog izvršenja vašeg programa.

Format ulaznih podataka je sljedeći: Prvo se pročita T (broj podtestova). Potom, za svaki podtest:

- linija 1: dva cijela broja - N and M ;
- linija $2 + i$ (za $0 \leq i < M$): dva cijela broja - A_i i B_i , koji predstavljaju dva čvora koje povezuje ivica i ($0 \leq A_i, B_i < N$).

Primjer ocjenjivača će potom ispisati broj različitih boja koje je vaše rješenje koristilo i broj iteracija koje su se izvršile prije nego što se robot ugasio. U slučaju da vaše rješenje nije bilo uspješno, ispisaće poruku sa greškom.

Primjer ocjenjivača će ispisivati detaljne informacije o tome šta robot vidi i radi u svakoj iteraciji. To možete onemogućiti ako promijenite vrijednost `DEBUG` iz `true` to `false`.