

Задача Навигација

 3 сек.  512 MB

Постои **поврзан ненасочен едноставен кактус-граф**¹ со $N \leq 1000$ јазли и M ребра. Неговите јазли имаат бои (означени со ненегативни цели броеви од 0 до 1499). Првично, сите јазли имаат боја 0. **Детерминистички робот без меморија**² го истражува графот движејќи се од јазол до јазол. Мора да ги посети сите јазли барем еднаш, а потоа да прекине.

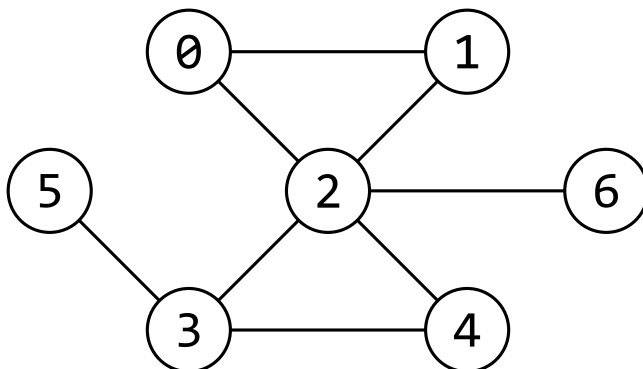
Роботот започнува од некој јазол, кој може да биде кој било од јазлите во графот. На секој чекор, ја гледа бојата на својот сегашен јазол и боите на сите соседни јазли **во некој редослед фиксиран за сегашниот јазол** (т.е. повторното посетување на јазолот ќе му даде на роботот иста низа на соседни јазли, дури и ако нивните бои се различни од претходно). Роботот прави едно од следниве две дејства:

1. Одлучува да прекине.
2. Избира нова (или можеби иста) боја за сегашниот јазол и на кој соседен јазол да се премести. Соседниот јазол се идентификува со индекс од 0 до $D - 1$, каде што D е бројот на соседни јазли.

Во вториот случај, сегашниот јазол се пребојува (или можеби останува иста боја) и роботот се преместува на избраниот соседен јазол. Ова се повторува сè додека роботот не прекине или додека не ја достигне границата на итерација. Роботот победува ако ги посети сите јазли, а потоа заврши во рамките на ограничувањето на итерацијата од $L = 3000$ чекори (во спротивно губи).

Треба да дизајнирате стратегија за роботот што може да го реши проблемот за кој било кактус-граф. Дополнително, треба да го минимизирате бројот на различни бои што ги користи вашето решение. Бојата 0 секогаш се смета за употребена.

¹Поврзан ненасочен едноставен кактус-граф е поврзан ненасочен едноставен граф (секој јазол е достапен од секој друг јазол; ребрата се двонасочни; нема самојамки или повеќекратни ребра) во кој секое ребро припаѓа на најмногу еден едноставен циклус (едноставен циклус е циклус што го содржи секој јазол најмногу еднаш). Сликата подолу е пример.





²Роботот е детерминистички и без меморија, ако неговото дејство зависи само од неговите сегашни влезни податоци (т.е. не складира податоци од чекор до чекор) и секогаш го избира истото дејство кога му се дадени истите влезни податоци.



Детали за имплементацијата

Стратегијата на роботот треба да се имплементира како следната функција:

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

Како параметри ги прима бојата на сегашниот јазол и боите на сите соседни јазли (по редослед). Мора да врати пар чиј прв елемент е новата боја за сегашниот јазол, а како втор елемент е индексот на соседниот јазол кон кој треба да се движи роботот. Ако наместо тоа, роботот треба да заврши, функцијата треба да го врати парот $(-1, -1)$.

Оваа функција ќе се повикува постојано за да се изберат дејствата на роботот. Бидејќи е детерминистичка, ако `navigate` веќе е повикана со некои параметри, никогаш повеќе нема да се повика со истите параметри; наместо тоа, нејзината претходна вратена вредност ќе биде повторно употребена. Дополнително, секој тест може да содржи $T \leq 5$ подтеста (различни графови и/или почетни позиции) и тие може да се извршуваат истовремено (т.е. вашата програма може да добива наизменични повици за различни подтести). Конечно, повиците до `navigate` може да се случат во **одделни извршувања** на вашата програма (но понекогаш може да се случат и во истото извршување). Вкупниот број на извршувања на вашата програма е $P = 100$. Поради сето ова, вашата програма не треба да се обидува да пренесува информации помеѓу различни повици.



Ограничувања

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

Scoring

Делот S од поените за подзадача што ќе го добиете зависи од C – максималниот број на различни бои што ги користи вашето решение (вклучувајќи ја и бојата 0) на кој било тест во таа подзадача или која било друга задолжителна подзадача:

- Ако вашето решение не успее на кој било подтест, тогаш $S = 0$.
- Ако $C \leq 4$, тогаш $S = 1.0$.
- Ако $4 < C \leq 8$, тогаш $S = 1.0 - 0.6 \frac{C-4}{4}$.
- Ако $8 < C \leq 21$, тогаш $S = 0.4 \frac{8}{C}$.
- Ако $C > 21$, тогаш $S = 0.15$.

Подзадачи

Подзадача	Поени	Задолжителни подзадачи	N	Дополнителни ограничувања
0	0	—	≤ 300	Примерот.
1	6	—	≤ 300	Графот е циклус. ¹
2	7	—	≤ 300	Графот е ѕвезда. ²
3	9	—	≤ 300	Графот е патека. ³
4	16	2 – 3	≤ 300	Графот е дрво. ⁴
5	27	—	≤ 300	Сите јазли имаат најмногу 3 соседни јазли, а јазолот од кој роботот започнува има 1 соседен јазол.
6	28	0 – 5	≤ 300	—
7	7	0 – 6	—	—

¹Графот на циклусот има рабови: $(i, (i + 1) \bmod N)$ за $0 \leq i < N$.

²Графот на ѕвездите има рабови: $(0, i)$ за $1 \leq i < N$.

³Графот на патеките има рабови: $(i, i + 1)$ за $0 \leq i < N - 1$.

⁴Дрвото е граф без циклуси.

Пример

Разгледајте го примерокот на графикон од сликата во исказот, кој има $N = 7$, $M = 8$ и рабови $(0, 1)$, $(1, 2)$, $(2, 0)$, $(2, 3)$, $(3, 4)$, $(4, 2)$, $(3, 5)$ и $(2, 6)$. Дополнително, бидејќи редоследот на елементите во листите на соседство на јазлите е релевантен, ги даваме во оваа табела:



Node	Соседни јазли
0	2, 1
1	2, 0
2	0, 3, 4, 6, 1
3	4, 5, 2
4	2, 3
5	3
6	2

Да претпоставиме дека роботот започнува на јазолот 5. Потоа, следново е една можна (неуспешна) низа на интеракции:

#	Colors	Node	Повик до <code>navigate</code>	Вратена вредност
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>

Тука роботот користел вкупно 6 различни бои: 0, 1, 2, 3, 4 и 8 (забележете дека 0 би се сметале за користени дури и ако роботот никогаш не ја вратил бојата 0, бидејќи сите јазли започнуваат во бојата 0). Роботот работел 9 итерации пред да заврши. Сепак, не успеал бидејќи завршил без да го посети јазолот 1.

¹Забележете дека повикот до `navigate` при итерација 4 всушност нема да се случи. Ова е затоа што е еквивалентен на повикот при итерација 1, па затоа оценувачот едноставно ќе ја употреби повратната вредност на вашата функција од тој повик. Сепак, ова сè уште се смета за итерација на роботот.



Sample grader

Пример оценувачот не извршува повеќекратни извршувања на вашата програма, па затоа сите повици до `navigate` ќе бидат во истото извршување на вашата програма.

Влезниот формат е следниот: Прво се чита T (бројот на подтести). Потоа за секој подтест:

- line 1: два цели броја - N и M ;
- line $2 + i$ (за $0 \leq i < M$): два цели броја - A_i и B_i , кои се двата јазли што ги поврзува работ i ($0 \leq A_i, B_i < N$).

Потоа, пример-оценувачот ќе го испечати бројот на различни бои што ги користело вашето решение и бројот на итерации што му биле потребни пред да заврши. Алтернативно, ќе испечати порака за грешка, ако вашето решение не успее.

Стандардно, пример-оценувачот печати детални информации за тоа што роботот гледа и прави при секоја итерација. Можете да го оневозможите ова, со промена на вредноста на `DEBUG` од `true` во `false`.