

Dizajniranje igračka

Naziv zadatka	Dizajniranje igračka
Datoteka za unos	interaktivni zadatak
Datoteka za ispis	interaktivni zadatak
Vremensko ograničenje	1 sekunda
Memorijsko ograničenje	256 megabajta

Radite u tvrtci koja dizajnira igračke. Novostvorena igračka radi ovako: Dano je n priključaka, označenih brojevima od 1 do n , koji vire van kutije. Neki parovi priključaka povezani su žicama unutar kutije. (Drugim riječima, priključci i žice tvore neusmjeren graf, gdje su priključci čvorovi te žice bridovi) Žice nisu vidljive izvana te jedini način za saznati nešto o njima je korištenje **ispitivača** na priključcima: Moguće je izabrati dva priključka i te j takva da $i \neq j$ te ispitivač će odrediti jesu li ta dva priključka spojena unutar kutije, izravno ili neizravno. (Drugim riječima, ispitivač će odrediti postoji li put između dva dana priključka u pripadnom grafu).

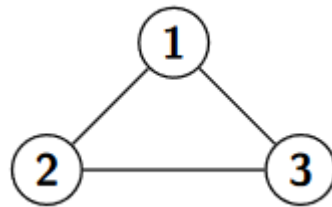
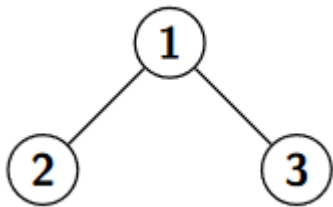
Skup veza unutar kutije nazivamo **dizajnom** igračke.

Koristite posebnu programsku opremu kako biste ispitivali te stvarala takve dizajne. Programska operma radi na sljedeći način: započinje s nekim dizajnom igračke koji nazivamo "dizajn 0". Veze dizajna unutar kutije se ne prikazuju. Umjesto toga, možete više puta ponoviti sljedeći postupak:

1. Odabrati broj dizajna a i dva priključka i te j takve da $i \neq j$.
2. Programska oprema prikazat će vam što bi se dogodilo kada bismo primjenili ispitivač na odabrani par. Drugim riječima, reći će vam jesu li priključci i te j (izravno ili neizravno) povezani unutar dizajna a .
3. Također, ako priključci i te j nisu bili izravno povezani u dizajnu a , stvara novi dizajn sa svim prijašnjim vezama gdje su dodatno priključci i te j izravno povezani. Tom dizajnu dana je sljedeća dostupna oznaka. (Prvom je dizajnu dana oznaka 1, drugom 2 i tako dalje...) Primjetite da to ne mijenja dizajn a već samo stvara novi dizajn.

Vaš je cilj saznati što više o dizajnu 0.

Primjetite da nije uvijek moguće saznati točan skup veza nad dizajnom 0 jer nije moguće razlikovati izravne i neizravne veze. Primjerice, promatrajte sljedeća dva dizajna s $n = 3$:



Ispitivač bi svaki par priključaka prepoznao kao spojene unutar oba dizajna, te ih stoga nije moguće razlikovati opisanom programskom opremom.

Vaš je cilj odrediti dizajn ekvivalentan dizajnu 0. Dva dizajna su **ekivalentna** ako ispitivač vraća isti rezultat unutar oba dizajna za svaki par priključaka.

Implementacija

Ovo je *interkativan zadatak*. Morate implementirati sljedeću funkciju

```
void ToyDesign(int n, int max_ops);
```

koja određuje dizajn koji je *ekivalentan* dizajnu 0. Vaša implementacija trebala bi to postići pozivanjem dviju određenih operacija nula ili više puta. U jednoj operaciji, program bi trebao pozvati funkciju:

```
int Connected(int a, int i, int j);
```

gdje $1 \leq i, j \leq n$, $i \neq j$, $a \geq 0$, i a nisu veći od trenutnog broja dizajna. Ako priključci i te j su (izravno ili neizravno) povezani u dizajnu a , onda će vratiti nazad broj a . Inače, vratiti će broj trenutnih dizajnova uvećan za jedan, što je i oznaka novog dizajna koji dijeli sve veze s dizajnom a te dodatno sadrži vezu između priključaka i te j . Funkcija `Connected` smije biti pozvana najviše `max_ops` puta.

Kada je vaš program gotov s `Connected` operacijama, trebao bi opisati dizajn ekvivalentan dizajnu 0 te vratiti. Kako bi opisali dizajn, program bi trebao pozvati:

```
void DescribeDesign(std::vector<std::pair<int,int>> result);
```

Parametar `result` je vektor parova brojeva koji opisuju veze među priključcima. Svaki par opisuje točno jednu vezu te trebao bi sadržavati dva priključka koja su njome međusobno spojena. Smije postojati najviše jedna veza za svaki neuređeni par priključaka, te ne smiju postojati veze koje spajaju priključak sa samim sobom.

Ograničenja

- $2 \leq n \leq 200$

Bodovanje

- Podzadatak 1 (10 bodova): $n \leq 200$, $max_ops = 20\ 000$
- Podzadatak 2 (20 bodova): $n \leq 8$, $max_ops = 20$
- Podzadatak 3 (35 bodova): $n \leq 200$, $max_ops = 2\ 000$
- Podzadatak 4 (35 bodova): $n \leq 200$, $max_ops = 1\ 350$

Primjer interakcije

Potez natjecatelja	Potez ocjenjivača	Objašnjenje
	<code>ToyDesign(4, 5)</code>	Unutar igračke nalaze se 4 priključka. Potrebno je odrediti dizajn ekvivalentan dizajnu 0 pozivajući funkciju <code>Connected</code> najviše 5 puta.
<code>Connect(0, 1, 2)</code>	Vraća 1.	Priključci 1 i 2 nisu povezani niti izravno niti neizravno unutar dizajna 0. Novi dizajn 1 je stvoren.
<code>Connected(1, 3, 2)</code>	Vraća 2.	Priključci 3 i 2 nisu spojeni niti izravno niti neizravno unutar dizajna 1. Stvoren je novi dizajn 2.
<code>Connected(0, 3, 4)</code>	Vraća 0.	Priključci 3 i 4 nisu spojeni niti izravno niti neizravno unutar dizajna 0. Ne stvara se novi dizajn.
<code>DescribeDesign({{3, 4}})</code>	-	Opisujemo dizajn s točno jednom vezvom: Između priključaka 3 i 4.

Primjer ocjenjivača

Dan vam je primjer ocjenjivača unutar `grader.cpp`, u prilogu zadatka `ToyDesign.zip` koji učitava unos iz standardnog unosa na sljedeći način:

- Prvi redak sadrži broj čvorova n , broj bridova m i max_ops
- Sljedećih m redaka sadrži parove čvorova kao parove.

Primjer ocjenjivača učitava unos i poziva `ToyDesign` funkciju iz natjecateljevog rješenja.

Kako biste sagradili svoje rješenje zajedno s ocjenjivačem, unutar naredbenog retka unesti sljedeću naredbu:

```
g++ -std=gnu++11 -O2 -o solution grader.cpp solution.cpp
```

gdje je `solution.cpp` vaše rješenje poslano na CMS. Kako biste pokrenuli svoj program zajedno s primjerom unosa danom u prilogu, unesite sljedeću naredbu u naredbeni redak:

```
./solution < input.txt
```