

## Žaislų dizainas (Toy Design)

Užduoties pavadinimas	Žaislų dizainas
Įvesties failas	Interaktyvi užduotis
Išvesties failas	Interaktyvi užduotis
Laiko ribojimas	1 sekundė
Atminties ribojimas	256 megabaitai

Jūs dirbate kompanijai, kuri kuria žaislus. Naujo žaislo kūrimas veikia taip: yra  $n$  kaištukų, sunumeruotų nuo 1 iki  $n$ , kurie išsikiša iš dėžės. Kai kurie kaištukai yra poromis sujungti laidais dėžės viduje. (Kitais žodžiais, kaištukai ir laidai sudaro nekryptinį grafą, kur kaištukai yra viršūnės ir laidai yra briaunos.) Laidai yra nematomi iš išorės, ir vienintelis būdas ką nors apie juos išsiaiškinti yra naudoti **testerį** ant kaištukų: mes galime pasirinkti du kaištukus  $i$  ir  $j$ , tokius, kad  $i \neq j$ , ir testeris pasakys, ar tie kaištukai yra sujungti dėžės viduje, arba tiesiogiai arba netiesiogiai. (Kitais žodžiais, testeris pasakys, ar yra kelias grafe tarp šių kaištukų.)

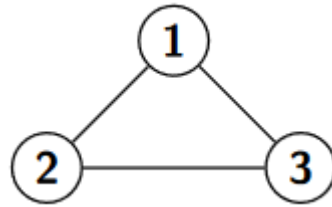
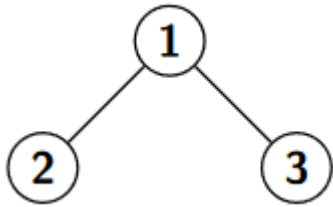
Šią viduje dėžės esančių sujungimų aibę vadinsime žaislo **dizainu**.

Jūs naudojate specializuotą programinę įrangą, kad užklaustumėte ir sukurtumėte šiuos dizainus. Ši programinė įranga veikia šitaip: ji pradeda su kažkokiu žaislo dizainu, kurį mes pažymėsime „dizainas 0“. Programinė įranga jums šio dizaino sujungimų dėžės viduje neparodo. Vietoje to, jūs galite kartoti šią trijų žingsnių operaciją:

1. Pasirenkate dizaino numerį  $a$  ir du kaištukų numerius  $i$  ir  $j$ , tokius, kad  $i \neq j$ .
2. Programinė įranga jums pasako, kas nutiktų, jei panaudotume testerį ant šių dviejų kaištukų. Kitais žodžiais, ji pasako, ar kaištukai  $i$  ir  $j$  yra (tiesiogiai arba netiesiogiai) sujungti dizaine  $a$ .
3. Taip pat, jei kaištukai nebuvo tiesiogiai arba netiesiogiai sujungti dizaine  $a$ , tada ji sukuria naują dizainą, kuris turi visus sujungimus iš dizaino  $a$  ir dar vieną tiesioginį sujungimą tarp  $i$  ir  $j$ . Šis dizainas yra pažymėtas kitu galimu dizaino skaičiumi. (Tad, pirmas dizainas sukurtas šiuo būdu turės skaičių 1, tada skaičių 2, ir taip toliau.) Atkreipkite dėmesį, kad tai nepakeičia dizaino  $a$ , tik sukuria naują dizainą, kuris turi papildomą jungtį.

Jūsų tikslas – sužinoti kiek įmanoma daugiau apie dizainą 0 naudojantis šia operacija.

Atkreipkite dėmesį, kad ne visada įmanoma tiksliai nustatyti dizaino 0 sujungimus, nes nėra būdo atskirti tiesioginį ir netiesioginį sujungimą. Pavyzdžiui, panagrinėkime šiuos du dizainus su  $n = 3$ :



Testeris pasakytų, kad bet kurie du kaištukai yra sujungti abiejuose dizainuose, tad mes negalėsime jų atskirti naudojant aukščiau aprašytą programinę įrangą.

Jūsų tikslas – nustatyti bet kokį dizainą, kuris yra lygiavertis dizainui 0. Du dizainai yra **lygiaverčiai**, jei testerio parodymai sutampa visoms kaištukų poroms abiejuose dizainuose.

## Realizacija

Ši užduotis yra interaktyvi. Jums reikia realizuoti funkciją

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

kuri nustato dizainą, lygiavertį dizainui 0. Jūsų realizacija turėtų pasiekti šį tikslą iškviesdama dvi funkcijas kaip aprašyta žemiau. Pirma funkcija, kurią galite iškviesiti, yra:

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

kur  $1 \leq i, j \leq n$ ,  $i \neq j$ ,  $a \geq 0$ , ir  $a$  turi neviršyti kol kas sukurtų dizainų kiekio. Jei kaištukai  $i$  ir  $j$  yra (tiesiogiai ar netiesiogiai) sujungti dizaine  $a$ , tada ši funkcija grąžins  $a$  atgal. Kitu atveju, grąžins jau sukurtų dizainų kiekį pridėjus vieną. Šis skaičius pataps naujo dizaino numeriu. Naujas dizainas turi visas jungtis iš dizaino  $a$ , pridėjus jungtį tarp  $i$  ir  $j$ . Funkcija `Connected` gali būti iškviesta ne daugiau nei `max_ops` kartų.

Kai jūsų programa baigė naudoti `Connected` operacijas, ji turėtų nupasakoti dizainą, kuris lygiavertis dizainui 0. Kad nupasakotų dizainą, programa turėtų iškviesiti:

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

Parametras `result` yra sveikųjų skaičių porų vektorius, aprašantis tiesiogines jungtis tarp kaištukų. Kiekviena pora atitinka vieną jungtį ir turėtų turėti du jungties kaištukų numerius. Tarp bet kurių dviejų kaištukų turi būti ne daugiau nei viena tiesioginė jungtis, ir negali būti jokios jungties kaištuko su savim pačiu. Šios funkcijos iškvietimas pabaigs jūsų programos darbą.

## Ribojimai

- $2 \leq n \leq 200$

## Vertinimas

- Dalinė užduotis 1 (10 taškų):  $n \leq 200$ ,  $max\_ops = 20\,000$
- Dalinė užduotis 2 (20 taškų):  $n \leq 8$ ,  $max\_ops = 20$
- Dalinė užduotis 3 (35 taškai):  $n \leq 200$ ,  $max\_ops = 2\,000$
- Dalinė užduotis 4 (35 taškai):  $n \leq 200$ ,  $max\_ops = 1\,350$

## Pavyzdinė komunikacija

Dalyvio veiksmas	Vertinimo programos veiksmas	Paaiškinimas
	<code>ToyDesign(4, 20)</code>	Žaisle yra 4 kištukai. Jums reikia surasti bet kokį dizainą, kuris lygiavertis dizainui 0, iškviečiant <code>Connected</code> ne daugiau nei 20 kartų.
<code>Connected(0, 1, 2)</code>	Gražina 1.	Kaištukai 1 ir 2 nėra tiesiogiai ar netiesiogiai sujungti dizaine 0. Sukurtas naujas dizainas 1.
<code>Connected(1, 3, 2)</code>	Gražina 2.	Kaištukai 3 ir 2 nėra tiesiogiai ar netiesiogiai sujungti dizaine 1. Sukurtas naujas dizainas 2.
<code>Connected(0, 3, 4)</code>	Gražina 0.	Kaištukai 3 ir 4 yra tiesiogiai ar netiesiogiai sujungti dizaine 0. Naujas dizainas nėra sukuriamas.
<code>DescribeDesign({{3, 4}})</code>	-	Aprašome dizainą, kuris turi tik vieną jungtį: tarp kaištukų 3 ir 4.

## Pavyzdinė vertinimo programa

Duotoji pavyzdinė vertinimo programa, `grader.cpp`, užduoties prisegtuke `ToyDesign.zip`, nuskaity įvestį iš standartinės įvesties tokiu formatu:

- Pirmoje eilutėje yra kaištukų skaičius  $n$ , tiesioginių jungčių kiekis  $m$  ir  $max\_ops$ .
- Kitose  $m$  eilučių yra aprašytos tiesioginės jungtys kaip kaištukų poros.

Pavyzdinė vertinimo programa nuskaity įvestį ir iškviečia `ToyDesign` funkciją, esančią dalyvio sprendime. Vertinimo programa išves vieną iš toliau aprašytų žinučių, pagal jūsų sprendimo

elgseną:

- „Wrong answer: Number of operations exceeds the limit.“ (Neteisingas atsakymas: operacijų kiekis viršija limitą), jei funkcijos `Connected` iškvietimų kiekis viršija `max_ops`;
- „Wrong answer: Wrong design id.“ (Neteisingas atsakymas: neteisingas dizaino ID), jei parametras `a` kviečiant funkciją `Connected` yra dizaino numeris, kuris tuo metu dar neegzistuoja;
- „Wrong answer: Incorrect design.“ (Neteisingas atsakymas: neteisingas dizainas), jei dizainas, aprašytas per `DescribeDesign`, nėra lygiavertis dizainui 0.

Kad sukompiliumėtumėte pavyzdinę vertinimo programą su savo sprendimu, jūs galite naudoti šią komandą terminale (komandinėje eilutėje):

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

kur `solution.cpp` yra jūsų sprendimų failas, kurį norite įkelti į CMS. Kad paleistumėte programą su pavyzdine įvestimi (pridėta užduoties prisegtuke), įveskite šią komandą į terminalą:

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