Chika nori sukčiauti (Chika Wants to Cheat)

Užduoties pavadinimas	Chika nori sukčiauti
Įvesties failas	Interaktyvi užduotis
Išvesties failas	Interaktyvi užduotis
Laiko ribojimas	2 sekundės
Atminties ribojimas	512 megabaitai

Chika turi q dydžio kortų kaladę, sunumeruotą įvairiais teigiamais sveikaisiais skaičiais. Ji nori su jomis pažaisti keletą žaidimų kartu su savo draugais iš Shuchi'in Akademijos studentų tarybos. Ji taip pat nori laimėti, todėl nusprendžia slapta pažymėti kortų nugarėles.

Visos kortos yra kvadrato formos ir 2×2 dydžio, kur apatinis kairysis kampas turi koordinates $(0, 0)$, ir viršutinis dešinysis kampas turi $(2, 2)$ koordinates. Chika nupiešia tam tikrą raštą ant kiekvienos kortos nugarėlės, kad ji vėliau, žiūrėdama į jį, žinotų, kuris skaičius užrašytas kortos priekyje. Ji piešia raštą tokia tvarka: tiek kartų, kiek ji nori (gali būti ir 0 kartų), ji pasirenka du skirtingus taškus A ir B, turinčius sveikąsias koordinates apatinio kairiojo kortos kampo atžvilgiu ir tarp jų nubrėžia **tiesią atkarpą**.

Chika brėš tik **tinkamas** atkarpas, tokias, kad toje pačioje atkarpoje tarp taškų A ir B neegzistuoja joks kitas taškas C su sveikųjų skaičių koordinatėmis (nesutampantis su A ir B). Pavyzdžiui, atkarpa tarp $(0, 0)$ ir $(2, 2)$ yra **netinkama**, nes ant jos yra taškas $(1, 1)$, bet atkarpos nuo $(0, 0)$ iki $(1, 1)$ ir nuo $(1, 1)$ iki $(2, 2)$ yra abi **tinkamos** ir Chika gali jas piešti nors ir kuriant tą patį raštą. Svarbu paminėti, jog linijos neturi krypties: linija nupiešta iš A į B yra **identiška** tiek pati sau, tiek linijai nupieštai iš B į A.

Svarbu tai, kad Chika nori būti tikra, kad atpažins savo kortas, nepaisant to, kaip jos yra pasuktos. Kortelę galima pasukti 0, 90, 180 arba 270 laipsnių prieš laikrodžio rodyklę nuo originalios padėties.

Jūsų užduotis yra padėti Chikai sukurti raštus visoms q kortoms ir vėliau atpažinti tas kortas.

Realizacija

*Ši užduotis yra interaktyvi su dviem etapais, **kiekvienas etapas paleis jūsų programą atskirai***. Jums reikia realizuoti dvi funkcijas:

- `BuildPattern` funkciją, kuri grąžina raštą, kurį reiktų nupiešti ant duotos kortos. Ši funkcija bus iškviečiama q kartų pirmame etape.
- `GetCardNumber` funkciją, kuri grąžina (galimai pasuktos) kortos skaičių, kuri turi duotą raštą, nupieštą pirmame etape. Ši funkcija bus iškviesta q kartų antrame etape.

Pirmoji funkcija

```
std::vector<std::pair<std::pair<int, int>, std::pair<int, int>>> BuildPattern(int n);
```

priima vieną parametrą n, skaičių, kuris yra užrašytas kortos priekinės pusės. Jums reikia grąžinti `std::vector` su atkarpomis, kurias Chika nupieš kaip raštą ant kortos nugarėlės, kad galėtų paskui ją pažinti. Atkarpa aprašoma kaip taškų pora `std::pair`, ir taškai aprašomi kaip sveikųjų koordinačių (x, y) pora `std::pair`, kurios skaičiuojamos nuo apatinio kairiojo kortelės kampo, kur $0 \leq x, y \leq 2$. Visos atkarpos, kurias Chika nupiešia, turi būti tinkamos ir skirtingos. Garantuojama, kad visi q iškvietimų gaus skirtingus n parametrus.

Kai gaus visus q kortų raštus, vertinimo programa gali padaryti vieną iš toliau aprašytų operacijų, bet kokį kiekį kartų, kiekvienam iš raštų:

- Pasukti visą raštą 0, 90, 180 arba 270 laipsnių prieš laikrodžio rodyklę.
- Pakeisti atkarpų tvarką rašto vektoriaus `std::vector` aprašyme.
- Pakeisti atkarpos galų tvarką rašte. (Atkarpa, nubrėžta iš A į B gali patapti vienoda su atkarpa iš B į A.)

Antroji funkcija, `int GetCardNumber(std::vector<std::pair<std::pair<int, int>, std::pair<int, int>>> p);`

priima vieną parametrą p, atkarpų vektorių `std::vector`, aprašantį Chikos nupieštą raštą ant kortos nugarėlės, paremtą pagal ankstesnio jūsų `BuildPattern` funkcijos iškvietimo grąžintą reikšmę. Funkcija turi grąžinti skaičių n, parašytą ant kortelės viršutinės pusės. Prisiminkite, kad raštas p nebūtinai yra originalia forma, kokią gražino `BuildPattern`; ji galėjo būti pakeista aukščiau aprašytomis trijomis operacijomis. Taip pat gali būti, kad kortų tvarka skiraisi nuo tos, kuria jos buvo paduotos pirmame etape, tačiau garantuojama, kad kiekviena kortelė bus panaudota lygiai vieną kartą.

Ribojimai

- $1 \leq q \leq 10\,000$.

- $1 \leq n \leq 67\,000\,000$ visiems funkcijos `BuildPattern` iškvietimams.
- Atkreipkite dėmesį, kad egzistuoja algoritmas sukontstruoti tokius raštus, kad galima atpažinti $67\,000\,000$ skirtingų kortų.

Vertinimas

- Dalinė užduotis 1 (2 taškai): $n \leq 2$.
- Dalinė užduotis 2 (9 taškai): $n \leq 25$.
- Dalinė užduotis 3 (15 taškų): $n \leq 1\,000$ ir vertinimo programa raštų **nepasuks** tarp 1 ir 2 etapo. (Vertinimo programa **gali** atlikti kitas dvi operacijas.)
- Dalinė užduotis 4 (3 taškai): $n \leq 16\,000\,000$ ir vertinimo programa raštų **nepasuks** tarp 1 ir 2 etapo. (Vertinimo programa **gali** atlikti kitas dvi operacijas.)
- Dalinė užduotis 5 (24 taškai): $n \leq 16\,000\,000$.
- Dalinė užduotis 6 (18 taškų): $n \leq 40\,000\,000$.
- Dalinė užduotis 7 (29 taškai): Jokių papildomų ribojimų.

Pavyzdinė komunikacija

Funkcijos iškvietimas	Grąžinama reikšmė	Paaiškinimas
Pirmas programos etapas prasideda.	-	-
`BuildPattern(3)`	`{{{0, 0}, {2, 1}}, {{1, 1}, {2, 0}}}`	Mums reikia sukurti raštą skaičiui 3 ant 2×2 dydžio kortelės. Mes nusprendžiame nubrėžti dvi atkarpas: - tarp $(0, 0)$ ir $(2, 1)$, - tarp $(1, 1)$ ir $(2, 0)$.
`BuildPattern(1)`	`{{{0, 1}, {0, 0}}}`	Mums reikia sukurti raštą skaičiui 1 ant 2×2 dydžio kortelės. Mes nusprendžiame nubrėžti vieną atkarpą: - tarp $(0, 1)$ ir $(0, 0)$.
Pirmas programos etapas baigiasi.	-	-
Antras programos etapas prasideda.	-	-
`GetCardNumber( {{{0, 0}, {0, 1}}})`	`1`	Mes gauname raštą, sudarytą iš 1 atkapos: - tarp $(0, 0)$ ir $(0, 1)$. Tai tas pats raštas, kaip ir būtume gavę nubrėžę atkarpą: - tarp $(0, 1)$ ir $(0, 0)$ kuris yra lygiai toks pats raštas su tokia pačia orientacija (pasukta 0 laipsnių),

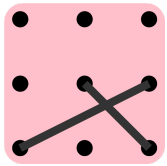
		kurį mes grąžinome po antro funkcijos `BuildPattern` iškvietimo. Todėl grąžiname 1.

`GetCardNumber(`
`{{{1, 1}, {2, 2}},`
`{{1, 2}, {2, 0}}})`
	`3`	Mes gauname raštą, sudarytą iš 2 atkapų:
		- tarp $(1, 1)$ ir $(2, 2)$,
		- tarp $(1, 2)$ ir $(2, 0)$.
		Šis raštas yra toks pats, kurį grąžinome po pirmo funkcijos `BuildPattern` iškvietimo, pasuktas 90 laipsnių prieš laikrodžio rodyklę. Tad grąžiname 3.

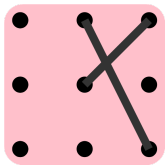
Antras programos etapas baigiasi.	-	-

Kiti trys paveikslėliai parodo, tokia tvarka:

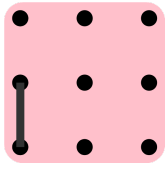
- Raštą, grąžintą pirma kartą iškvietus funkciją `BuildPattern`:



- Raštą, gautą kaip parametrą antrą kartą kveičiant funkciją `GetCardNumber`, kuris yra pirmasis raštas, tik pasuktas 90 laipsnių prieš laikrodžio rodyklę.



- Raštą, grąžintą iškvietus `BuildPattern` antrą kartą, kuris yra tas pats raštas, gautas kaip parametras iškviečiant `GetCardNumber` pirmą kartą.



Pavyzdinė vertinimo programa

Duotoji pavyzdinė vertinimo programa, `grader.cpp`, užduoties prisegtuke `Cheat.zip`, nuskaito sveikąjį skaičių q iš standartinės įvesties ir tada kartoja šiuos žingsnius q kartų:

- Nuskaito sveikąjį skaičių n iš standartinės įvesties.
- Iškviečia `BuildPattern(n)` ir išsaugo gautą reikšmę kitamajame p.

- Iškviečia `GetCardNumber(p)` ir išveda grąžintą reikšmę į standartinę išvestį.

Jūs galite keisti savo vertinimo programos kopiją lokaliai, jei norite taip padaryti.

Kad sukompiliuotumėte 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
```

Atkreipkite dėmesį, kad, ne taip kaip vertinimo programa, tikra vertinimo programa esanti CMS įvykdys pirmą ir antrą etapus per skirtingus jūsų programos paleidimus.