

SLara Vara

Problem Name	SLara Vara
Datoteka za unos	interaktivni zadatak
Datoteka za ispis	interaktivni zadatak
Vremensko ograničenje	2 sekunde
Memorijsko ograničenje	512 megabajta

SLara ima svežanj od q karata za igranje označenih raznim prirodnim brojevima. SLara se želi igrati sa svojim prijateljicama iz MIOC-a, no također želi i pobijediti te je stoga odlučila tajno označiti stražnje strane karata.

Karte su kvadratnog oblika i veličine 2×2 , gdje je donji lijevi kut označen koordinatama $(0, 0)$ i gornji desni $(2, 2)$. SLara će nacrtati određeni uzorak na stražnjoj strani svake karte, kako bi kasnije na temelju uzorka mogla prepoznati o kojoj se karti radi. Crtat će uzorke sljedećim postupkom: Koliko god puta htjela (moguće i 0 puta), odabrat će dvije različite točke A i B s cjelobrojnim koordinatama u odnosu na donji lijevi kut te nacrtati **ravnu dužinu** među njima.

SLara će crtati isključivo **valjane** dužine, točnije, dužine među točkama A i B za koje ne postoji točka C (različita od A i B) s cjelobrojnim koordinatama koja se također nalazi na dužini. Primjerice, dužina između točke $(0, 0)$ i točke $(2, 2)$ **nije valjana** jer sadrži točku $(1, 1)$, ali dužine među točkama $(0, 0)$ i $(1, 1)$ te $(1, 1)$ i $(2, 2)$ jesu **valjanje**, i SLara može čak nacrtati obje unutar istog uzorka. Također, primjetite da su dužine neusmjerene, dužina nacrtana od točke A do točke B identična je dužini nacrtanoj od točke B do točke A .

Također, SLara mora biti sposobna prepoznati uzorak neovisno o rotaciji karte. Karta može biti rotirana za 0 , 90 , 180 ili 270 stupnjeva u smjeru suprotnom od kazaljke na satu s obzirom na izvornu usmjerenje karte.

Vaš je zadatak pomoći SLari stvoriti uzorke za q karata njezinog svežnja te joj također kasnije pomoći prepoznati iste.

Implementacija

Ovo je interaktivan zadatak gdje se interakcija odvija u dva čina, **svaki čin uključuje zasebno pokretanje Vašeg programa**. Potrebno je implementirati sljedeće dvije funkcije:

- `BuildPattern` je funkcija koja vraća uzorak na začelju pojedine karte. Ova funkcija biti će pozvana q puta unutar prvog čina.
- `GetCardNumber` je funkcija koja vraća oznaku pojedine karte na temelju uzorka stvorenog u prvom činu koji je možda i zarotiran. Ova funkcija biti će pozvana q puta unutar drugog čina.

Prva funkcija

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

prima jedan parametar n , broj napisan na prednjoj strani karte. Potrebno je vratiti `std::vector` koje je SLara nacrtala na začelju na temelju kojih će ju kasnije prepoznati. Dužina je prikazana kao `std::pair` točaka, a točka kao `std::pair` (x, y) cjelobrojnih koordinata u odnosu na donji lijevi kut karte, gdje vrijedi $0 \leq x, y \leq 2$. Sve dužine koje SLara nacrtala moraju biti valjane i neidentične.

Nakon što je primio svi q uzoraka, ocjenjivač smije napraviti sljedeći postupak nad bilo kojim uzorkom proizvoljan broj puta:

- Zarotirati cijeli uzorak za 0, 90, 180 ili 270 stupnjeva u smjeru suprotnom od kazaljke na satu.
- Promijeniti redoslijed dužina unutar `std::vector` prikaza uzorka.
- Promijeniti redoslijed krajnjih točaka dužine. (Dužina između točaka A i B može postati identična dužina između točaka B i A)

Druga funkcija,

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

prima jedan parametar p , `std::vector` dužina koje opisuju uzorak nacrtan na stranžnjoj strani, određene na temelju poziva Vaše funkcije `BuildPattern`. Funkcija mora vratiti broj n , oznaku karte napisanu na prednjoj strani. Prisjetite se da uzorak p nije nužno u izvornom obliku kakvog ga je vratila funkcija `BuildPattern`; možda su na njega primijenjene tri gore spomenuta postupka. Također je moguće da je redoslijed kojim su uzorci dani različit od redoslijeda karata tijekom prvog čina.

Ograničenja

- $1 \leq q \leq 10\,000$.
- $1 \leq n \leq 67\,000\,000$ za sve pozive funkcije `BuildPattern`.

- Primjetite da postoji algoritam koji može razaznati 67 000 000 različitih karata.

Bodovanje

- Podzadatak 1 (2 bodova): $n \leq 2$.
- Podzadatak 2 (9 bodova): $n \leq 25$.
- Podzadatak 3 (15 bodova): $n \leq 1000$ i ocjenjivač **neće rotirati** uzorke između prvog i drugog čina. (Ocjenjivač će **možda** primjenjivati druga dva postupka.)
- Podzadatak 4 (3 bodova): $n \leq 16\,000\,000$ i ocjenjivač **neće rotirati** uzorke između prvog i drugog čina. (Ocjenjivač će **možda** primjenjivati druga dva postupka.)
- Podzadatak 5 (24 bodova): $n \leq 16\,000\,000$.
- Podzadatak 6 (18 bodova): $n \leq 40\,000\,000$.
- Podzadatak 7 (29 bodova): Nema dodatnih ograničenja.

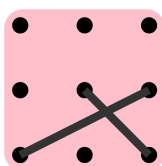
Primjer interakcije

Poziv funkcije	Vraćena vrijednost	Objašnjenje
Početak prvog pokretanja.	-	-
BuildPattern(3)	<code>{{{0, 0}, {2, 1}}, {1, 1}, {2, 0}}</code>	Moramo stvoriti uzorak za kartu oznake 3 na 2×2 karti. Odlučili smo nacrtati dvije dužine - između (0, 0) i (2, 1), - između (1, 1) i (2, 0).
BuildPattern(1)	<code>{{{0, 1}, {0, 0}}}</code>	Moramo stvoriti uzorak za kartu oznake 1 na 2×2 karti. Odlučili smo nacrtati jednu dužinu - između (0, 1) i (0, 0),
Završetak prvog pokretanja.	-	-
Početak drugog pokretanja.	-	-
GetCardNumber(<code>{{{0, 0}, {0, 1}}}</code>)	1	Primili smo uzorak sačinjen od jedne dužine - između (0, 0) i (0, 1). To je isti uzorak kao i onaj dobiven crtanjem dužine: - između (0, 1) i (0, 0) što je isti uzorak (zarotiran za nula stupnjeva) dobiven pozivom funkcije BuildPattern. Stoga vraćamo 11.
GetCardNumber(<code>{{{0, 0}, {0, 1}}}</code>)	3	Primili smo uzorak sačinjen od jedne

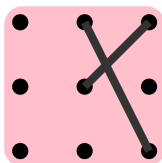
<pre>{{{1, 1}, {2, 2}}, {{1, 2}, {2, 0}}}}</pre>		dužine - između (1, 1) i (2, 2). - između (1, 2) i (2, 0) što je isti uzorak (zarotiran za 90 stupnjeva) dobiven pozivom funkcije <code>BuildPattern</code> . Stoga vraćamo 3.	
Završetak pokretanja.	drugog	-	-

Sljedeće tri slike redom prikazuju:

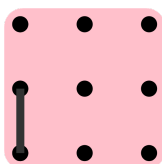
- Uzorak prvog poziva funkcije `BuildPattern`:



- Uzorak primljen tijekom drugog poziva funkcije `GetCardNumber`, koji je prvi uzorak zarotiran za 90 stupnjeva u smjeru suprotnom od smjera kazaljke na satu.



- Uzorak drugog poziva funkcije `BuildPattern`, koji je isti kao uzorak primljen tijekom prvog poziva funkcije `GetCardNumber`.



Primjer ocjenjivača

Dani primjer ocjenjivača `grader.cpp` u prilogu zadatka `Cheat.zip` čita broj iz standardnog unosa q i ponavlja sljedeće korake q puta.

- Učitavanje broja n iz standardnog unosa.
- Pozivanje `BuildPattern(n)` i spremanje dobivene vrijednosti u varijablu p .
- Pozivanje `GetCardNumber(p)` i ispis dobivene vrijednosti u standardni ispis.

Smijete lokalno mijenjati ocjenjivač

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
```

Primjetite da za razliku od primjera ocjenjivača, pravi će ocjenivač odvojeno pokretati funkcije unutar prvog i drugo čina na gore opisan način unutar dva pokretanja programa.