

Софија сака да мами

Име на задачата	Cheat
Влезна датотека	Интерактивна задача
Излезна датотека	Интерактивна задача
Временско ограничување	2 секунди
Мемориско ограничување	512 мегабајти

Софија има шпил со q карти за играње, нумерирани со различни позитивни цели броеви. Таа сака да игра некои игри со своите пријателки од македонскиот тим на EGOI 2022 користејќи ги овие карти, но исто така сака и да победува, па одлучува скришно да ги обележи картите од нејзиниот шпил на задната страна.

Сите карти се со квадратна форма и со големина 2×2 , кадешто долниот лев агол има координати $(0, 0)$, а горниот десен агол има координати $(2, 2)$. Софија црта некој облик на задната страна од секоја карта, за подоцна да дознае, штом ќе го погледне обликот, кој број е запишан на предната страна од картата. Овој облик таа го црта користејќи ја следната процедура: Колку што сака пати (може да се и 0 пати), таа избира две различни точки A и B коишто имаат целобројни координати релативно во однос на долниот лев агол на картата и црта **отсечка** помеѓу нив.

Софија ќе црта само **валидни** отсечки. Валидна е онаа отсечка помеѓу две точки A и B за којашто не постои друга точка C (различна од A и B) со целобројни координати која исто така лежи на отсечката. На пример, отсечката помеѓу точките $(0, 0)$ и $(2, 2)$ е **невалидна** бидејќи ја содржи точката $(1, 1)$, но двете отсечки: онаа помеѓу точките $(0, 0)$ и $(1, 1)$, како и онаа помеѓу точките $(1, 1)$ и $(2, 2)$, се **валидни** отсечки, и притоа Софија може да ги нацрта и двете во рамките на еден ист облик. Исто така, да забележиме дека отсечките немаат насока: отсечка нацртана од A до B е **идентична** сама со себе и со отсечката нацртана во обратната насока, од B до A .

Многу важно: Софија сака да се осигури дека ќе ги препознае нејзините карти независно од тоа како се ротирани. Една карта може да се заротира за 0 , 90 , 180 или 270 степени, спротивно од насоката на движење на стрелките на часовникот, во однос на нејзината оригинална ориентација.

Ваша задача е да и' помогнете на Софија да ги дизајнира облиците за q -те карти од нејзиниот шпил и потоа да ги препознае овие карти подоцна.

Имплементација

Ова е интерактивна задача со две фази, **при што секоја фаза вклучува одделно извршување на вашата програма**. Треба да имплементирате две функции:

- Функција `BuildPattern` којашто го враќа обликот што треба да се нацрта на задната страна на дадена карта. Оваа функција ќе биде повикана q пати во првата фаза.
- Функција `GetCardNumber` којашто го враќа бројот на картата (може да е заротирана) којашто поседува даден облик нацртан во првата фаза. Оваа функција ќе биде повикана q пати во втората фаза.

Првата функција

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

прима еден параметар n , бројот којшто е запишан на предната страна на картата. Треба да вратите `std::vector` што ги содржи отсечките што ги црта Софија како облик на задната страна на картата, за да го препознае подоцна. Отсечката е претставена како `std::pair` од точки, а точката е претставена како `std::pair` (x, y) од целобројни координати релативно во однос на долниот лев агол од картата, каде $0 \leq x, y \leq 2$. Сите отсечки што ги црта Софија мора да се валидни и да не се меѓусебно идентични. Се гарантира дека сите q повици до `BuildPattern` примаат различни вредности за параметарот n .

По примањето на сите облици за q -те карти, оценувачот може да ги изведе кои било од следниве операции, произволен број пати, над секој од облиците:

- Ротирај го целиот облик за $0, 90, 180$ или 270 степени спротивно од насоката на движење на стрелките на часовникот.
- Промени го редоследот на отсечките во `std::vector` репрезентацијата на обликот.
- Промени го редоследот на крајните точки на некоја отсечка во обликот (Отсечка нацртана од A до B може се смени во идентичната отсечка од B до A .)

Втората функција,

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

прима еден параметар p , `std::vector` од отсечки што го опишуваат обликот нацртан од страна на Софија на задната страна од картата, врз основа на повратната вредност од претходен повик до вашата `BuildPattern` функција. Функцијата мора да го врати бројот n запишан на предната страна од картата. Да се потсетиме дека обликот p не мора да биде во

оригиналната форма којашто е вратена од страна на функцијата `BuildPattern`; тој може да бил предмет на трите операции спомнати претходно. Возможно е, исто така, редоследот на картите да е различен од оној редослед по којшто биле зададени во првата фаза, но се гарантира дека секоја карта ќе биде употребена точно еднаш.

Ограничувања

- $1 \leq q \leq 10\,000$.
- $1 \leq n \leq 67\,000\,000$ за сите повици до функцијата `BuildPattern`.
- Да забележиме дека постојат алгоритми за конструкција на облици такви што да може да се препознаат $67\,000\,000$ различни карти.

Оценување

- Подзадача 1 (2 поени): $n \leq 2$.
- Подзадача 2 (9 поени): $n \leq 25$.
- Подзадача 3 (15 поени): $n \leq 1\,000$ и оценувачот **нема да ги ротира** облиците помеѓу фазите 1 и 2. (Оценувачот **може** да ги изведе останатите две операции.)
- Подзадача 4 (3 поени): $n \leq 16\,000\,000$ и оценувачот **нема да ги ротира** облиците помеѓу фазите 1 и 2. (Оценувачот **може** да ги изведе останатите две операции.)
- Подзадача 5 (24 поени): $n \leq 16\,000\,000$.
- Подзадача 6 (18 поени): $n \leq 40\,000\,000$.
- Подзадача 7 (29 поени): Нема дополнителни ограничувања.

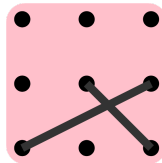
Пример интеракција

Функциски повик	Повратна вредност	Објаснување
Започнува првата фаза.	-	-
<code>BuildPattern(3)</code>	<code>{{{0, 0}, {2, 1}}, {{1, 1}, {2, 0}}}</code>	Треба да креираме облик за бројот 3 на картата со големина 2×2 . Одлучуваме да нацртаме 2 отсечки: - помеѓу (0, 0) и (2, 1), - помеѓу (1, 1) и (2, 0).
<code>BuildPattern(1)</code>	<code>{{{0, 1}, {0, 0}}}</code>	Треба да креираме облик за бројот 1 на картата со големина 2×2 . Одлучуваме да нацртаме 1 отсечка: - помеѓу (0, 1) и (0, 0).
Завршува првата фаза.	-	-
Започнува втората фаза.	-	-
<code>GetCardNumber({{{0, 0}, {0, 1}}})</code>	1	Добиваме облик составен од 1 отсечка:

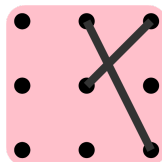
		<p>- помеѓу (0, 0) и (0, 1). Ова е истиот облик со оној што би го добиле со цртање на отсечката: - помеѓу (0, 1) и (0, 0) којшто е точно истиот облик со истата ориентација (заротиран за 0 степени) што го вративме при вториот повик на функцијата BuildPattern. Според тоа, враќаме 1.</p>
<pre>GetCardNumber({{1, 1}, {2, 2}}, {{1, 2}, {2, 0}})</pre>	3	<p>Добиваме облик составен од 2 отсечки: - помеѓу (1, 1) и (2, 2), - помеѓу (1, 2) и (2, 0). Ова е обликот што го вративме при првиот повик на функцијата BuildPattern, заротиран за 90 степени, спротивно од насоката на движење на стрелките на часовникот. Според тоа, враќаме 3.</p>
Завршува втората фаза.	-	-

Следните три слики ги претставуваат, по редослед:

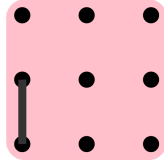
- Обликот вратен како излез од првиот повик до функцијата BuildPattern:



- Обликот примен како параметар за вториот повик до функцијата GetCardNumber, којшто е првиот облик но откако истиот е заротиран за 90 степени спротивно од насоката на движење на стрелките на часовникот.



- Обликот вратен како излез од вториот повик до функцијата BuildPattern, којшто исто така е истиот облик што е примен како аргумент за првиот повик до функцијата GetCardNumber.



Пример-оценувач

Дадениот пример-оценувач, `grader.cpp`, во додатокот на задачата `Cheat.zip`, чита еден цел број q од стандардниот влез и потоа ќе ги направи следните чекори q пати:

- Чита еден цел број n од стандардниот влез.
- Ја повикува `BuildPattern(n)` и ја зачувува повратната вредност во променлива p .
- Ја повикува `GetCardNumber(p)` и ја печати повратната вредност на стандарден излез.

Можете, доколку сакате, да го промените (модифицирате) вашиот оценувач локално.

За да го компајлирате пример-оценувачот со вашето решение, може да ја примените следната команда во командната линија (terminal prompt):

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

кадешто `solution.cpp` е датотеката со вашето решение кое ќе го прикачувате на CMS. За да ја извршите програмата со пример-влезот даден во додатокот, напишете ја следната команда во командната линија (terminal prompt):

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

Ве молиме забележете дека, за разлика од пример-оценувачот, вистинскиот оценувач на CMS ќе ги изведува првата и втората фаза во одделни извршувања на вашата програма.