



ТРЕНИРОВЪЧНО СЪСТЕЗАНИЕ НА РАЗШИРЕНИЯ НАЦИОНАЛЕН ОТБОР

София, 16 юли 2024 г.

Група А

Задача АТ13. МАГИЯ 2

🕒 1 сек. 📁 1024 МВ

Известният магьосник Хари е измислил следващия култов фокус, този път без асистентки. Започва се с тесте с N квадратни карти, номерирани с числата от 1 до N и се извиква някой зрител от публиката, който избира Q от картите. Той ги разбърква и започва да ги показва една по една, обърнати с гърба си към магьосника. Хари успява да познае карта след карта и макар в началото публиката да е захласната и впечатлена, то по някое време уличават Хари в измама.

Зрителят, който показва картите забелязва, че на гърбовете на картите има различни дребни маркировки с нарисувани прави отсечки. По-точно, оказва се, че на всяка от картите има район с размери 2 на 2, където са отсечките. Въвеждаме за определеност координатна система в този район, при която долният ляв ъгъл има координати $(0, 0)$, а горният десен ъгъл - $(2, 2)$. Отсечките, които са начертани са между 2 различни точки с целочислени координати. А дори зрителят забелязва и свойството, че на всяка отсечка има само 2 точки с целочислени координати (двата ѝ края). Такива отсечки ще наричаме *валидни*.

Магьосникът Хари е много хитър и дори е предвидил, че картите могат да са завъртени спрямо ориентацията на картата, при която е чертана маркировката. Въпреки това неговата магия (измама) работи безотказно.

Задача

Вие отново искате да докажете, че магия не съществува и затова ще напишете програма **magic2**, която да извършва описания фокус като изпълнява ролята на магьосника Хари. Вашата програма трябва да чертае маркировки по описания начин на зададени Q карти и след това да отгатва номерата на картите по начертаните маркировки.

Детайли по имплементацията

Трябва да имплементирате две функции (съответно за чертане на маркировка по номер на карта и за отгатване на номер на карта по начертана маркировка), които са в следния формат:

```
std::vector <std::pair <std::pair <int, int>, std::pair <int, int>>> mark_card (int k);  
int tell_card (std::vector <std::pair <std::pair <int, int>, std::pair <int, int>>> card);
```

Функцията `mark_card` ще бъде извикана Q пъти от програмата на журито с един параметър - номер на карта (число между 1 и N). Гарантирано е, че в рамките на един тест функцията се извиква за Q различни карти. Функцията трябва да върне **непразен** вектор със списъка на отсечките, които се използват за маркировката. Всяка отсечка представяме като наредена двойка от точки с целочислени координати (числа между 0 и 2), а всяка точка е представена като наредена двойка от цели числа. Отсечките трябва да спазват описанието за *валидни* отсечки и да не съвпадат една с друга.

Функцията `tell_card` ще бъде извикана Q пъти от програмата на журито с един параметър - описание на направената маркировка на някоя карта. Този параметър е вектор от отсечки, както върнатата стойност на функцията `mark_card`. Гарантирано е, че в рамките на един тест функцията се извиква за маркировките на Q -те карти, които са били подадени на `mark_card`, в същия ред. Функцията трябва да върне едно число, което е номерът на отгатнатата карта.

Програмата ви ще обработи Q заявки за карти, като първо ще бъде извикана функцията `mark_card` за всяка карта. След това за всяка маркировка на карта програмата на журито може:



ТРЕНИРОВЪЧНО СЪСТЕЗАНИЕ НА РАЗШИРЕНИЯ НАЦИОНАЛЕН ОТБОР

София, 16 юли 2024 г.

Група А

- да я завърти на ъгъл 90, 180 или 270 градуса по посока обратна на часовниковата стрелка;
- да промени реда на отсечките във вектора с отсечки;
- за всяка отсечка да промени реда на наредената двойка точки.

Така се гарантира, че вашата програма няма да кодира информация в реда на точки и отсечки, а в самите избрани отсечки. Освен това извикванията на първата функция ще са в различно изпълнение на вашата програма от извикванията на втората функция, за да се гарантира, че двете функции няма да споделят данни помежду си.

Вашата програма **magic2** трябва да имплементира двете описани функции. Тя може да съдържа и друг код, функции и глобални променливи, необходими за работата Ви, но не трябва да съдържа главната функция `main`. Също така, не трябва да четете от стандартния вход или да отпечатвате на стандартния изход.

Ограничения

- $1 \leq N \leq 6,7 \cdot 10^7$;
- $1 \leq Q \leq 10^4$.

Подзадачи

Подзадача	Точки	Необходими подзадачи	N	Други ограничения
1	2	—	≤ 2	—
2	9	1	≤ 25	—
3	15	—	$\leq 10^3$	Програмата на журито няма да завърта маркировки, но може да извършва останалите две действия.
4	3	3	$\leq 1,6 \cdot 10^7$	Програмата на журито няма да завърта маркировки, но може да извършва останалите две действия.
5	24	1 – 4	$\leq 1,6 \cdot 10^7$	—
6	18	1 – 5	$\leq 4 \cdot 10^7$	—
7	29	1 – 6	$\leq 6,7 \cdot 10^7$	—

Точките за дадена подзадача се получават само ако се преминат **успешно** всички тестове, предвидени за нея и необходимите подзадачи. Един тест е успешен, ако сте спазили протокола за комуникация и накрая сте отгатнали правилно всички карти.

Примерна комуникация

В примерната комуникация има две заявки, т.е. $Q = 2$.



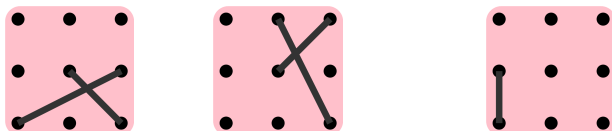
ТРЕНИРОВЪЧНО СЪСТЕЗАНИЕ НА РАЗШИРЕНИЯ НАЦИОНАЛЕН ОТБОР

София, 16 юли 2024 г.

Група А

Действия на вашата програма	Действия на журито
	<code>mark_card(3)</code>
<code>return {{{0, 0}, {2, 1}}, {{1, 1}, {2, 0}}}</code>	
	<code>mark_card(1)</code>
<code>return {{{0, 1}, {0, 0}}}</code>	
	Програмата на журито завърта първата маркировка на 90 градуса по посока обратна на часовниковата стрелка и променя реда на отсечките и реда на точките в описанието на отсечките.
	<code>tell_card({{{0, 0}, {0, 1}}})</code>
<code>return 1</code>	
	<code>tell_card({{{1, 1}, {2, 2}}, {{1, 2}, {2, 0}}})</code>
<code>return 3</code>	

Следващите фигури илюстрират различните маркировки. Първата фигура показва първата направена маркировка (за карта номер 3), а втората фигура показва подадената маркировка за познаване на карта номер 3 (след завъртането на 90 градуса от програмата на журито). Последната фигура показва втората направена маркировка за карта номер 1.



Локално тестване

За локално тестване са предоставени файловете `Lgrader.cpp` и примерен файл `magic2.cpp`, в който може да напишете вашата програма. Когато предоставените файлове са в една папка, може да компилирате заедно вашата програма `magic2.cpp` и `Lgrader.cpp`. Така ще получите програма, с която можете да проверите верността на функцията `Vi`.

Програмата ще изисква от стандартния вход следната последователност от данни:

- на първия ред: едно цяло положително число Q – броя карти за маркиране и отгатване;
- на следващите редове: едно цяло положително число k – номер на карта.

За всяка карта, програмата ще намери маркировката, която сте направили, след което може да се извършат действията, които и програмата на журито би направила с маркировката, и накрая ще се подаде получената маркировка за отгатване. Ако отгатването не е успешно за дадена карта, то ще получите подходящо съобщение за грешка. Иначе, ще получите съобщение "Correctly guessed card k .", където k е номерът на картата.

Ако искате маркировките да не се завъртат, то може да смените стойността на макроса `ROTATE` на `false`. Ако искате да не се извеждат маркировките, то може да смените стойността на макроса `DETAILED` на `false`.