

Содержание

0math 2

numberone 4

2manywires 6

3friends 8

4trees 11

0math

М. просто обожає математичні приколи. Цього разу її цікавлять «легендарні» числа - то є такі числа, які можна подати у вигляді $a^b \cdot b^a$, де a та b - **різні** цілі числа. У М. нещодавно зламався калькулятор, а в інформатиці вона не надто сильна, тож просить вас допомогти їй з перевіркою чергового числа на «легендарність».

Формат вхідних даних

В єдиному рядку ціле число N , що не перевищує 10^6 за абсолютною величиною - число для перевірки.

Формат вихідних даних

У разі, якщо N не можна подати у шуканому вигляді, виведіть «0 0» (без лапок). Інакше виведіть два різних цілих числа a та b (через пробіл), які за модулем не повинні перевищувати 10^6 .

Контрольний приклад 1	
72	2 3
Контрольний приклад 2	
30375	5 3

Пояснення до прикладів:

У першому варіанті маємо $72 = 2^3 \cdot 3^2$. Зверніть увагу, що можна було вивести пару «3 2» замість пари «2 3».

У другому варіанті $30375 = 3^5 \cdot 5^3$.

Ідея розв'язку задачі «0math»

Неважко побачити, що деяке $N \neq 1$ можна представити у вигляді $N^1 \cdot 1^N$. Однак для одиниці це невірно, адже a і b повинні бути різними. Однак для одиниці є інший варіант - $2^{-2} \cdot (-2)^2$.

numberone

А. позичила улюблене число свого старшого брата для гри. А. грає наступним чином: вона розрізає число на окремі цифри, після чого ставить між парами сусідніх цифр або «+», або ж «*». А. любить маленькі (іграшкові) речі, а тому і в результаті хоче отримати мінімальне число. Вона вже зіграла в гру і хоче впевнитися, чи мінімальний в неї результат, тож просить у вас знайти правильну відповідь.

Формат вхідних даних

В єдиному рядку ціле невід'ємне число N ($N \leq 20000$).

Формат вхідних даних

Виведіть єдине число - відповідь на задачу.

Контрольний приклад 1	
328	13
Контрольний приклад 2	
228	12
Контрольний приклад 3	
12	2

Пояснення до прикладів:

Усього чотири варіанти для контрольного прикладу №1:

- $3 \cdot 2 \cdot 8 = 48$
- $3 \cdot 2 + 8 = 14$
- $3 + 2 \cdot 8 = 19$
- $3 + 2 + 8 = 13$

Звісно, що у останньому варіанті результуюче число виявляється мінімальним

Для контрольного прикладу №2 оптимальним є варіант $2 \cdot 2 + 8$.

У третьому прикладі оптимальним є $1 \cdot 2$.

Ідея розв'язку задачі «numberone»

У випадку, якщо у записі числа присутній значущий нуль, можна результат перетворити в нуль (множенням усіх цифр).

У випадку, якщо повністю складається з одиниць, мінімальне число, що можна отримати - 1.

У інших випадках нам необхідно знайти суму **неодиничних** цифр числа.

2manywires

Поки користувацькі пристрої активно позбавляються від провідних технологій (останнім часом навіть у питаннях заряджання мобільних пристроїв), в серверних все ще використовуються дешеві, надійні та швидкі дротяні технології. Одна біда з ними - якщо кілька проводів заплутаються, знайти, де який починається і де закінчується перетворюється на той ще квест.

Саме з цією проблемою до вас звернувся усім відомий системний адміністратор Ю. Він має схему щитка, у якому дроти йдуть зверху до низу, якимось чином, можливо, переплітаючись. Є і гарна новина - щиток можна уявно розбити сіткою розмірами $N \times M$ сантиметрів. У такому випадку через жодну комірку не може пройти більше двох дротів. Вам необхідно віднайти для кожного вихідного дрота його номер серед вхідних (під вихідними дротами ми маємо на увазі ті, що розташовані у самому низу щитка, а під вхідними - дроти з самого верху). Дроти ми нумеруємо зліва направо.

Формат вхідних даних

В першому рядку задані розміри N, M ($2 \leq N, M \leq 200$). Далі слідують N рядків, в кожному з яких по M символів, що задають мапу дротів.

Символ «.» означає, що в даній комірці жодних дротів не проходить. Символ / Означає, що дріт йде вниз і ліворуч, відповідно, означає дріт, що йде вниз і праворуч. Символи «(» та «)» задають «перегин» дроту, коли він змінює напрямок. Крім того, ще існує символ «X», що означає перетин дротів в поточній комірці. Гарантується, що в першому і останньому рядках схеми щитка відсутні символи «X», «(», «)».

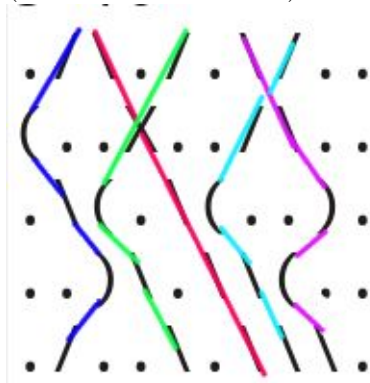
Формат вихідних даних

В першому рядку виведіть кількість дротів, що присутні на схемі. У другому рядку для кожного «виходу» виведіть номер «входу», що йому відповідає.

Контрольний приклад 1	
5 10 ./\./.\./.. (..X..\./.. .\(. \(. ..\)\.\.\(./..\.\.\.\.	5 1 3 2 5 4

Пояснення до прикладу:

(замість тисячі слів)



Ідея розв'язку задачі «2manywires»

Спершу пробіжимося по першому рядку «схеми» та пронумеруємо проводи. Заодно дізнаємося їх кількість.

Після цього пройдемо останнім рядком. Якщо зустріли символ «\» або «/» - значить, це кінець проводу. Будемо рухатися по проводу догори, починаючи з умовних «х» та «у». Запам'ятаємо напрямок (в залежності від того, який символ, це буде або «ліво», або ж «право»). Після цього будемо додавати чи віднімати 1 від поточної у-координати в залежності від поточного напрямку. Якщо наткнулися на символ «(» чи «)» - значить, необхідно змінити напрям. Символи «\», «/» та «X» не змінюють напрямку.

3friends

А., В. та М. вирішили створити свій стартап. Сучасних інвесторів складно зацікавити нейронними мережами чи блокчейном, тож замість цього хлопці вирішили створити аеропорт для квадрокоптерів. Для цього молодики знайшли ділянку поблизу столиці фонтанів розміром N на N метрів. Бідні студенти живуть лише на стипендію, тож змушені купувати землю по одному квадратному метру щомісяця. Щоб максимально ефективно використовувати бюджет, хлопці скористались останніми розробками в нейронних мережах та передбачили ціни на найближчий час. Після цього вони склали графік закупівлі землі.

Усім відомо, що найголовніша частина аеропорту — це злітна смуга. Міжнародні стандарти вимагають, щоб злітна смуга мала форму прямокутника, що має розмір A на B . Для полегшення роботи авіадиспетчера злітна смуга буде розміщена паралельно одній з сторін.

Як тільки буде побудована злітна смуга, можна починати наступний раунд інвестицій і запрошувати міжнародних експертів. Їх цікавить наступне питання: через скільки місяців буде куплена земельна ділянка, достатня для побудови злітної смуги. Звісно, що будувати студентам дозволено лише на куплених ділянках.

Писати нейронні мережі хлопці навчилися, а розв'язувати алгоритмічні задачі не дуже, тому й просять вас допомогти.

Формат вхідних даних

В першому рядку числа N, A, B ($1 \leq N \leq 1000, 1 \leq A, B \leq N$ - розміри. В наступних N рядках рівно по N різних натуральних чисел, кожне не перевищує N^2 . Кожне число задає номер місяця, у якому буде придбана відповідна ділянка землі.

Формат вихідних даних

Виведіть єдине число - мінімальний номер місяця, на якому з'явиться місце для побудови злітної смуги.

Контрольний приклад 1										
5	2	3								
3	13	14	15	18						
20	11	7	10	16						
12	2	4	8	17						
5	9	6	19	1						
21	24	25	23	22						

Пояснення до прикладу:

На момент 10 місяця «мапа» куплених земельних ділянок буде виглядати наступним чином (символ X позначає придбані ділянки, . - ще не придбані):

X
.	.	X	X	.
.	X	X	X	.
X	X	X	.	X

На 11 місяці купується ділянка:

X
.	X	X	X	.
.	X	X	X	.
X	X	X	.	X

Після чого злітну смугу розміру 2×3 можна розмістити двома способами.

Ідея розв'язку задачі «3friends»

Застосуємо ідею бінарного пошуку. Для певного моменту часу X перевіримо, чи можна розмістити злітну смугу на куплених ділянках. Для цього замінимо кожну клітинку на 0, якщо вона буде придбана після X -вого місяця або ж на 1 у іншому випадку. Тепер достатньо перевірити, чи існує одиничний підпрямокутник розміру $A \times B$. Варіантів зробити це ціла купа.

Асимптотика $O(N^2 \cdot \log N)$.

Інша ідея з такою самою асимптотикою наступна: давайте поставимо у відповідність позиції (x, y) прямокутник, лівий верхній кут якого розміщений у цій позиції. У такому разі нас цікавить підпрямокутник $(x..x + A - 1, y..y + B - 1)$, а точніше **максимум на підпрямокутнику** (тобто клітинку, яка з'явиться найпізніше). Узявши до уваги той факт, що максимум усього підпрямокутника - це максимум серед максимумів у кожному рядку, можемо за допомогою структури даних «дерево відрізків» отримати рішення за $O(N^2 \log N)$.

+ Існує рішення за $O(N^2)$, можна виділити на нього від 5 до 10 балів зі 100.

4trees

Робот-провідник Sousanin призначений для роботи у особливо дрімливих лісах Поділля. Нещодавно у ньому знайшли проблему - у деяких ситуаціях робот може заблудитися в чотирьох соснах. Співробітники лабораторії ЛІКТ вже зрозуміли алгоритм, за яким робот обходить перешкоди: натикаючись на дерево, робот повертає на 90 градусів праворуч, після чого йде прямолінійно до наступної зустрічі з деревом (або поки не вийде з лісу).

На жаль, виправлення програмного забезпечення - справа довготривала і дорога, тож туристичний клуб Вінниці вирішив знайти усі проблемні маршрути і видати відповідні брошури з рекомендаціями туристам. Проблемним маршрутом назовемо замкнуту послідовність обходу роботом чотирьох дерев, при цьому робот жодним чином (діючи за алгоритмом) вийти з цього маршруту не може.

Вам, як юному таланту Поділля, доручається розробити програму, яка б за малою дерев визначила б кількість проблемних маршрутів. Маршрути називаються різними, якщо відрізняється четвірка дерев, що в

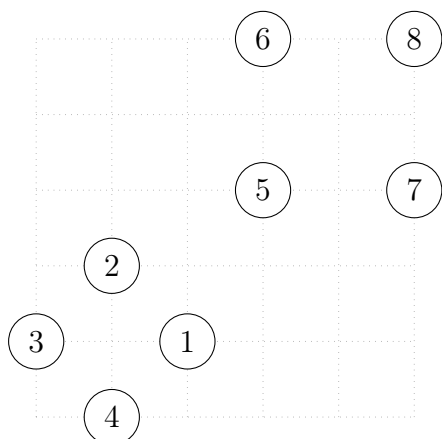
Формат вхідних даних

В першому рядку ціле число $N (1 \leq N \leq 1000)$ - кількість дерев на мапі. В наступних N рядках координати дерев у форматі $x_i y_i$. При цьому кожна з координат не перевищує 10^9 за абсолютною величиною. Жодних два дерева не розміщені в одній точці. Жодних три дерева не розміщені на одній прямій.

Формат вихідних даних

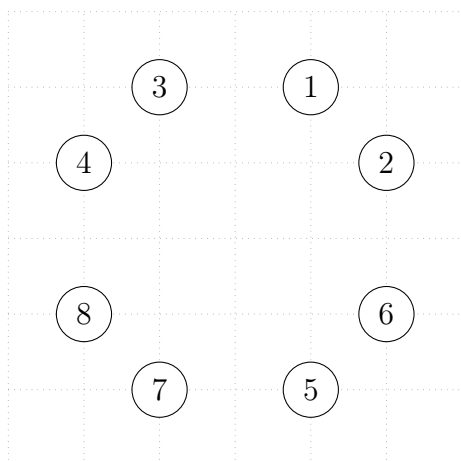
Виведіть єдине число - кількість проблемних маршрутів.

Контрольний приклад 1	
8 1 0 0 1 -1 0 0 -1 2 2 2 4 4 2 4 4	2
Контрольний приклад 2	
8 1 2 2 1 -1 2 -2 1 1 -2 2 -1 -1 -2 -2 -1	6

Пояснення до прикладів:

Проблемні маршрути наступні:

- $3 \rightarrow 2 \rightarrow 1 \rightarrow 4 \rightarrow 3$,
- $5 \rightarrow 6 \rightarrow 8 \rightarrow 7 \rightarrow 5$,



Проблемні маршрути наступні:

- $3 \rightarrow 1 \rightarrow 5 \rightarrow 7 \rightarrow 3$,
- $3 \rightarrow 2 \rightarrow 5 \rightarrow 8 \rightarrow 3$,
- $3 \rightarrow 6 \rightarrow 5 \rightarrow 4 \rightarrow 3$,
- $1 \rightarrow 2 \rightarrow 7 \rightarrow 8 \rightarrow 1$,
- $1 \rightarrow 6 \rightarrow 7 \rightarrow 4 \rightarrow 1$,
- $2 \rightarrow 6 \rightarrow 8 \rightarrow 4 \rightarrow 2$,

Ідея розв'язку задачі «4trees»

По суті, в задачі необхідно знайти кількість прямокутників, які можна побудувати на точках з деякої множини.

Згадаємо, що у прямокутнику діагоналі перетинаються у власних центрах. Крім того, самі діагоналі мають однакову довжину. Тому якщо ми хочемо перевірити, чи точки A, B, C, D утворюють прямокутник (саме у такому порядку), необхідно усього-навсього перевірити, чи однакові довжини відрізків AC та BD , а також чи середина відрізка AC співпадає з серединою відрізка BD . Усього у нас відрізків $\frac{N \cdot (N-1)}{2}$. Кожен відрізок ми можемо характеризувати трьома цілими числами - квадратом довжини відрізка та подвоєними координатами середини відрізка. В результаті нам необхідно перевірити кількість співпадінь таких трійок, що легко робиться за допомогою сортування і одного проходження по відсортованому масиву.

В результаті асимптотика $O(N^2 \cdot \log N)$.

За використання хеш-таблиць можна досягти асимптотики біля $O(N^2)$. На таке рішення можна відвести 5-10 додаткових балів.