

Загальна інформація

Загальне

На тренувальному турі вам дані задачі, щоб ви ознайомились з системою тестування. Результати тренувального туру **ніяк** не впливають на ваш результат на олімпіаді. Не зважаючи на це, ми **наполегливо рекомендуємо** вам розв'язати як можна більше задач. Задачі А-В надані для ознайомлення з системою тестування. Задачі С-Е — приклади легких задач, які використовувались на схожих змаганнях. Задачі F-J — це перший тур III етапу Всеукраїнської олімпіади з інформатики 2019.

Пам'ятка

Пам'ятка опублікована за посиланням <http://oi.in.ua/wp-content/uploads/2020/01/pamiatka.pdf>

Інструкція по користуванню системою

Опублікована інструкція по користуванню системою ejudge
<https://oi.in.ua/wp-content/uploads/2020/01/ejudge-quide.pdf>

Результати

За результатами офіційних турів можна буде стежити онлайн на сайті <https://oi.in.ua/> Проте учасники зможуть побачити результати лише після завершення кожного тура.

Підготовка

Рекомендуємо використати наступний список тем для підготовки <http://bit.ly/inf-res-20>

Зворотний зв'язок

Будь-які питання можна ставити на форумі <https://oi.in.ua/forum/uoi-3-2020>

Задача А. Сума

Назва вхідного файлу: `aplusb.in`
Назва вихідного файлу: `aplusb.out`
Обмеження використання часу: 1 second
Обмеження використання пам'яті: 256 megabytes

Потрібно знайти суму чисел a та b .

Формат вхідних даних

Перший рядок містить два цілі числа a та b ($-10^{18} \leq a, b \leq 10^{18}$).

Формат вихідних даних

Виведіть суму двох даних чисел.

Приклади

<code>aplusb.in</code>	<code>aplusb.out</code>
10 6	16
-56 42	-14

Оцінювання

Кожний тест, крім вхідних, оцінюється в 5 балів.

Приклади рішень

Приклад на C++:

```
#include <bits/stdc++.h>
using namespace std;
int main(){
    freopen("aplusb.in", "r", stdin);
    freopen("aplusb.out", "w", stdout);
    long long a, b;
    cin >> a >> b;
    cout << a + b << endl;
}
```

Приклад на Python 3:

```
import sys
sys.stdin = open('aplusb.in', 'r')
sys.stdout = open('aplusb.out', 'w')
a, b = map(int, input().split())
print(a + b)
```

Приклад на Java:

```
import java.io.*;
import java.util.*;
public class Main {
    public static void main(String[] args) throws IOException {
        Scanner sc = new Scanner(new File("aplusb.in"));
        PrintWriter pw = new PrintWriter(new File("aplusb.out"));
        long n = sc.nextLong();
        long m = sc.nextLong();
        pw.println(n + m);
        pw.close();
    }
}
```

Приклад на Pascal:

```
Var a, b: int64;
```

```
Begin
```

```
  Assign(input, 'aplusb.in'); Reset(input);
```

```
  Assign(output, 'aplusb.out'); Rewrite(output);
```

```
  Read(a,b);
```

```
  Writeln(a+b);
```

```
End.
```

Задача В. Сума чисел масиву

Назва вхідного файлу:	array-sum.in
Назва вихідного файлу:	array-sum.out
Обмеження використання часу:	1 second
Обмеження використання пам'яті:	256 megabytes

Дано n цілих чисел a_1, a_2, \dots, a_n . Знайдіть їх суму.

Зверніть увагу, що в цій задачі вам потрібно спробувати різні методи зчитування даних. Журі гарантує, що існують рішення на 100 балів на таких мовах: C++, Python 3, Java, Pascal. Якщо ви отримуєте менше 100 балів, то ви використовуєте не найшвидший метод зчитування.

Формат вхідних даних

Перший рядок містить одне ціле число n ($1 \leq n \leq 2 \cdot 10^6$) — кількість чисел.

Другий рядок містить n цілих чисел a_1, a_2, \dots, a_n ($0 \leq a_i \leq 10^9$) — числа масиву.

Формат вихідних даних

Виведіть одне число — суму чисел масиву.

Приклад

array-sum.in	array-sum.out
5 4 1 8 1 1	15

1. (10 балів) $n \leq 1\,000$, $a_i \leq 1\,000$
2. (10 балів) $n \leq 10\,000$
3. (8 балів) $n \leq 200\,000$
4. (8 балів) $n \leq 400\,000$
5. (8 балів) $n \leq 600\,000$
6. (8 балів) $n \leq 800\,000$
7. (8 балів) $n \leq 1\,000\,000$
8. (8 балів) $n \leq 1\,200\,000$
9. (8 балів) $n \leq 1\,400\,000$
10. (8 балів) $n \leq 1\,600\,000$
11. (8 балів) $n \leq 1\,800\,000$
12. (8 балів) Без додаткових обмежень.

Задача С. Клас

Назва вхідного файлу:	class.in
Назва вихідного файлу:	class.out
Обмеження використання часу:	1 second
Обмеження використання пам'яті:	256 megabytes

Клас являє собою матрицю $n \times m$. Тобто всього $n \cdot m$ парт, за кожною з яких сидить рівно один студент.

Час змін! Усі вони хочуть змінити свої місця. Якщо студент сидить за партою (x, y) , то він хоче пересісти на одну з парт: $(x + 1, y)$, $(x, y + 1)$, $(x - 1, y)$, $(x, y - 1)$. Якщо певної парті немає, то туди пересісти неможливо.

Вам потрібно визначити чи можуть усі студенти пересісти так, як вони хочуть.

Формат вхідних даних

Перший рядок містить одне ціле число t ($1 \leq t \leq 1000$) — кількість тестів.

Кожний з наступних t рядків містить два цілі числа n та m ($1 \leq n, m \leq 10^5$) — розміри класу.

Формат вихідних даних

Для кожного тесту вам потрібно вивести «ТАК», якщо студенти можуть пересісти так, як вони хочуть, інакше виведіть «НІ».

Зверніть увагу, що вам потрібно виводити букви латинського алфавіту, а не кирилицького.

Приклад

class.in	class.out
3	ТАК
2 2	НІ
1 1	ТАК
1 2	

Примітка

У першому прикладі студенти, які сидять в одному ряду, можуть поміняти місцями.

У другому прикладі всього один студент, який не може нікуди пересісти.

У третьому прикладі два студенти можуть поміняти місцями.

Задача D. Футбол

Назва вхідного файлу:	football.in
Назва вихідного файлу:	football.out
Обмеження використання часу:	0.5 seconds
Обмеження використання пам'яті:	256 megabytes

Микола Вікторович прийшов на футбольний матч. Як справжній футбольний фанат, він записував рахунок гри на листочку після кожного забитого голу. Наприклад, на листочку рахунки могли бути записані у такому порядку 0:1, 1:1, 2:1, 3:1, 3:2.

Матч був досить цікавим, а голів було дуже багато, тому Миколі Вікторовичу набридло записувати всі рахунки гри. Але він ще є хорошим математиком, тому він запам'ятав суму всіх чисел у рахунках, які повинні були бути записані.

Вам Микола Вікторович, як своєму другу, сказав лиш цю суму, яку він запам'ятав, а от рахунки гри він забув. Та якщо ви просто скажете йому кількість забитих голів, його влаштує й це. Ваше завдання — знайти цю кількість голів, або повідомити, що Микола Вікторович не міг отримати таку суму і заплутався у розрахунках.

Формат вхідних даних

Перший рядок містить одне ціле число x ($1 \leq x \leq 10^9$) — суму, яку запам'ятав Микола Вікторович.

Формат вихідних даних

Якщо існує можлива кількість голів — виведіть цю кількість. В іншому випадку виведіть -1 .

Приклади

football.in	football.out
6	3
1	1
5	-1

Примітка

У першому прикладі можливі такі рахунки: 0:1, 0:2, 1:2. Забитих голів — 3. Сума чисел — 6.

У другому прикладі рахунок може бути або 0:1, або 1:0.

У третьому прикладі неможливо отримати таке число.

Задача Е. Парне чи непарне

Назва вхідного файлу: `even-or-odd.in`
Назва вихідного файлу: `even-or-odd.out`
Обмеження використання часу: 0.5 seconds
Обмеження використання пам'яті: 256 megabytes

Перевірте чи кількість дільників числа непарна.

Формат вхідних даних

Перший рядок містить одне ціле число n ($1 \leq n \leq 100$) — кількість чисел.

Другий рядок містить n цілих чисел a_i ($1 \leq a_i \leq 10^{18}$) — числа, парність кількості дільників яких потрібно перевірити.

Формат вихідних даних

Для кожного числа виведіть 1, якщо кількість дільників непарна, або 0, якщо парна.

Приклад

<code>even-or-odd.in</code>	<code>even-or-odd.out</code>
4 1 3 6 4	1001

Примітка

Число 1 має лише один дільник — 1.

Число 3 має два дільники: 1 та 3.

Число 6 має чотири дільники: 1, 2, 3 та 6.

Число 4 має три дільники: 1, 2 та 4.

Оцінювання

Ви отримаєте бали за блок лише у випадку проходження усіх тестів блока.

- (20 балів) $1 \leq a_i \leq 10^5$
- (30 балів) $1 \leq a_i \leq 10^9$
- (50 балів) Без додаткових обмежень.

Задача F. Козак Вус і важлива знахідка

Назва вхідного файлу:	lesson.in
Назва вихідного файлу:	lesson.out
Обмеження використання часу:	0.25 seconds
Обмеження використання пам'яті:	256 megabytes

Зовсім нещодавно мешканці Потоколяндії знайшли стародавню табличку розміром 2×2 , в якій розташовано чотири числа A , B , C і D так:

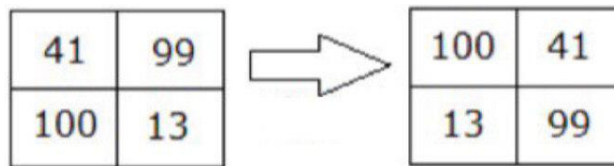
A	B
C	D

Вони відразу зрозуміли, що це дуже важлива історична знахідка. Спершу вони віднесли її Козаку Вусу для того, щоб він визначив важливість цієї таблички. На думку Козака Вуса, важливість таблички дорівнює $A \cdot (B + C - D)$.

На жаль, правильне положення таблички невідоме. Тому може статись таке, що однозначно визначити важливість таблички неможливо, оскільки це значення залежить від того, скільки разів її обернути.

Припустимо, що один оберт — це оберт за годинниковою стрілкою на 90° .

Наприклад, якщо $A = 41$, $B = 99$, $C = 100$, $D = 13$, то важливість рівна $41 \cdot (99 + 100 - 13) = 7626$. А якщо її один раз обернути, то $100 \cdot (41 + 13 - 99) = -4500$.



Козак Вус хоче з'ясувати, яку максимальну можливу важливість може мати ця знахідка. Але Вас він просить дізнатись, яку мінімальну кількість обертів потрібно зробити для того, щоб важливість таблички була максимальною.

Формат вхідних даних

Перший рядок містить чотири цілих числа A , B , C і D ($-10^8 \leq A, B, C, D \leq 10^8$) — числа на табличці.

Формат вихідних даних

Виведіть мінімальну кількість обертів, які потрібно зробити Козаку Вусу, щоб важливість таблички була максимальною.

Приклади

lesson.in	lesson.out
5 3 4 6	1
2 9 -4 13	3
2 6 3 0	0

Примітка

У першому прикладі спочатку табличка має важливість 5, але, якщо обернути її один раз, то вона набуде максимальної важливості, що рівна 32.

У другому прикладі достатньо трьох обертів, щоб табличка отримала своє максимальне значення важливості, яке рівне 171.

У останньому прикладі табличку повертати не треба, бо вона вже має максимальне значення важливості, яке рівне 18.

Оцінювання

Кожний тест, крім вхідних, оцінюється в 5 балів.

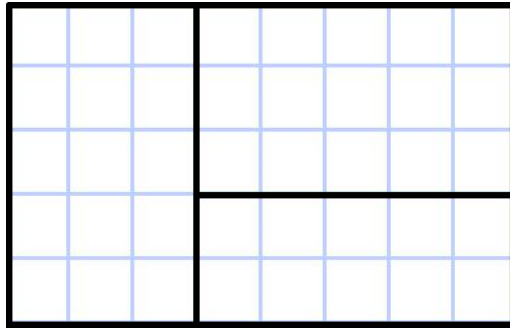
Задача G. Козак Вус і цікава задача

Назва вхідного файлу:	room.in
Назва вихідного файлу:	room.out
Обмеження використання часу:	0.25 seconds
Обмеження використання пам'яті:	256 megabytes

Всім відомо, що Козак Вус дуже захоплюється математикою. Сьогодні, читаючи книгу «Конкретна математика», він знайшов дуже цікаву задачу і вирішив запропонувати Вам її розв'язати.

Є кімната, яка має прямокутну форму. Одна з її сторін має довжину n , а друга — m . Вам необхідно з'ясувати, чи можливо розділити кімнату на **рівно три** окремі кімнати з цілими довжинами сторін так, щоб загальний периметр цих кімнат був рівно p .

Наприклад, якщо $n = 5$, $m = 8$ і $p = 46$, то один з можливих варіантів поділу кімнати такий:



Виведіть «YES» та розміри кімнат, якщо кімнату можливо розділити на три окремі кімнати, відповідно до умови задачі, інакше — «NO».

Формат вхідних даних

Перший рядок містить три цілих числа n , m та p ($1 \leq n, m \leq 10^9, 1 \leq p \leq 10^{15}$) — довжини сторін та периметр відповідно.

Формат вихідних даних

Виведіть «YES», якщо кімнату можливо розділити на три окремі кімнати з цілими довжинами сторін так, щоб загальний периметр цих кімнат був рівний p , інакше — «NO».

Якщо відповідь «YES», тоді у кожному з наступних трьох рядків треба вивести розміри відповідної кімнати. Розміри можна виводити у будь-якому порядку. Якщо відповідей декілька, то виведіть будь-яку.

Приклади

room.in	room.out
2 2 14	YES 2 1 1 1 1 1
2 3 17	NO
5 8 46	YES 5 3 2 5 3 5

Примітка

У першому прикладі можна поділити кімнату на три кімнати з розмірами 2×1 , 1×1 та 1×1 відповідно.

У другому прикладі неможливо поділити кімнату на три із цілими сторонами, так щоб загальний периметр був рівний 17.

Третій приклад пояснено в умові.

Оцінювання

Кожний тест, крім прикладів, оцінюється від 1 до 4 балів.

Задача Н. Козак Вус і НСД

Назва вхідного файлу:	gcdarray.in
Назва вихідного файлу:	gcdarray.out
Обмеження використання часу:	0.5 seconds
Обмеження використання пам'яті:	256 megabytes

Сьогодні Козак Вус зустрівся зі своїм давнім другом — Козаком Вухом. Вони дуже довго розмовляли, згадували своє дитинство та юність. Так і зайшла мова про задачу, яку колись вони не змогли вирішити на олімпіаді з програмування.

Дано масив з n чисел. За один хід можна одне з чисел **збільшити на 1**. Вам необхідно з'ясувати, за яку мінімальну кількість операцій можливо отримати масив, який буде задовольняти такі умови:

- $a_i \leq a_{i+1}$ для всіх i від 1 до $n - 1$.
- найбільший спільний дільник усіх чисел більший за 1.

Найбільший спільний дільник множини додатних чисел — це найбільше додатне число, що одночасно є дільником усіх чисел з множини.

Формат вхідних даних

Перший рядок містить одне ціле число t ($1 \leq t \leq 5$) — кількість тестів. Далі слідує опис кожного тесту.

Перший рядок опису кожного тесту містить одне ціле число n ($1 \leq n \leq 10^4$) — розмір масиву.

Другий рядок опису кожного тесту містить n цілих чисел a_1, a_2, \dots, a_n ($1 \leq a_i \leq 10^4$) — числа масиву.

Формат вихідних даних

Для кожного тесту в окремому рядку виведіть одне число — мінімальну кількість операцій, які необхідно виконати для того, щоб масив задовольняв дані умови.

Приклади

gcdarray.in	gcdarray.out
1 3 9 1 16	10
2 4 5 7 3 6 5 4 2 8 16 10	7 8

Примітка

У першому прикладі можна перше та друге число збільшити до 10, тоді найбільший спільний дільник чисел з масиву буде рівний два.

У першому тесті другого прикладу можна усі числа зробити рівними 7.

У другому тесті другого прикладу масив можна змінити до масиву [4, 4, 8, 16, 16].

Оцінювання

№	Обмеження	Додаткові обмеження	Бали
1	$1 \leq n \leq 10^4$	Усі числа парні	10
2	$1 \leq n \leq 10$	-	20
3	$1 \leq n \leq 10^3$	-	30
4	$1 \leq n \leq 10^4$	-	40

Задача I. Козак Вус і Потоколяндія

Назва вхідного файлу:	roads.in
Назва вихідного файлу:	roads.out
Обмеження використання часу:	2 seconds
Обмеження використання пам'яті:	512 megabytes

У Потоколяндії є n будинків, у i -му з яких проживають a_i мешканців. Між цими будинками є m доріг, кожна дорога сполучає будинки v_i і u_i . Ми визначаємо щастя кожного мешканця як кількість мешканців (включно з собою), яких він може зустріти. Житель будинку може зустріти іншого жителя, якщо він з його будинку, або з будинку до якого можна дістатися, подорожуючи по дорогах Потоколяндії.

Останні d днів кожного дня відбувалася подія одного з двох типів:

1. Снігом засипало дорогу між будинками g_i і h_i , тож тепер по ній не можна проїхати.
2. k_i мешканців w_i -го будинку на вертольоті відправлялися у гості до далеких родичів за межами Потоколяндії.

Мешканці Потоколяндії пишуть листи Козаку Вусу з проханням повідомити їм останній такий день, що сума щастя будь-якого мешканця x_i -го будинку та будь-якого мешканця y_i -го будинку принаймні z_i .

Можна вважати, що всі дії відбуваються миттєво в першу мить кожного дня. Якщо і до початку всіх подій сумарне щастя менше за z_i , то потрібно вивести -1 . Якщо сумарне щастя було не менше за z_i лише до першої події, то потрібно вивести 0 . Якщо після i -ої події сумарне щастя стало меншим, ніж потрібно, то потрібно вивести $i - 1$. Якщо після всіх подій сумарне щастя принаймні z_i , то потрібно вивести d .

Оскільки Козак Вус досить зайнята людина, а мешканців Потоколяндії багато, він просить Вас допомогти йому відповісти на всі листи.

Формат вхідних даних

Перший рядок містить чотири цілих числа n , m , d та s ($1 \leq n, m, d, s \leq 2 \cdot 10^5$) — кількість будинків, доріг, днів та повідомлень відповідно.

Наступний рядок містить n цілих чисел a_1, a_2, \dots, a_n ($1 \leq a_i \leq 10^9$) — кількість жителів в i -му будинку.

Кожний з наступних m рядків містить по два цілих числа v_i та u_i ($1 \leq v_i, u_i \leq n$, $u_i \neq v_i$) — будинки між якими є дорога. Гарантується, що немає кратних ребер.

Кожен з наступних d рядків описує запит в одному з двох форматів:

- «1 g_i h_i » ($1 \leq g_i, h_i \leq n$) — дорога, яку засипало снігом. Гарантується, що така дорога існує, і вона не була засипана снігом раніше.
- «2 w_i k_i » ($1 \leq w_i \leq n$, $1 \leq k_i \leq 10^9$) — номер будинку та кількість людей, які покинули його. Гарантується, що в будь-який момент часу в кожному будинку буде принаймні один мешканець.

Кожен з наступних s рядків містить по три цілих числа x_i , y_i та z_i ($1 \leq x_i, y_i \leq n$, $1 \leq z_i \leq 10^9$).

Формат вихідних даних

Виведіть s окремих рядків, у кожному з яких відповідь на відповідний запит. Якщо сума щастя двох мешканців ще до першого дня менша z , то виведіть -1 .

Приклади

roads.in	roads.out
3 3 5 4 2 9 4 1 2 2 3 1 3 2 2 1 1 2 3 2 3 3 1 1 2 1 1 3 1 2 11 3 2 20 1 3 15 2 3 10	4 3 3 4
4 5 3 4 1 2 3 4 1 2 2 3 1 3 3 4 2 4 1 2 4 1 3 4 2 3 2 1 4 21 1 4 20 1 3 9 2 2 2	-1 1 2 3

Примітка

Пояснення другого прикладу:

Для першого запиту сумарне щастя двох мешканців завжди буде менше 21 (спочатку воно рівне 20). Для другого запиту, після другого дня їх сумарне щастя буде зменшено до $6 + 4 = 10$. Інші запити пояснюються так само.

Оцінювання

№	Обмеження			Додаткові	Бали
	n, m	d	s		
1	$1 \leq n, m \leq 200$	$1 \leq d \leq 200$	$1 \leq s \leq 200$	-	4
2	$1 \leq n, m \leq 2000$	$1 \leq d \leq 2000$	$1 \leq s \leq 2000$		7
3	$1 \leq n, m \leq 2 \cdot 10^5$				13
4	$1 \leq n, m \leq 5000$	$1 \leq d \leq 5000$	14		
5	$1 \leq n, m \leq 2 \cdot 10^5$	$1 \leq d \leq 2 \cdot 10^5$	$1 \leq s \leq 2 \cdot 10^5$	$x_i = y_i$	27
6				-	35

Задача J. Козак Вус і свята

Назва вхідного файлу:	events.in
Назва вихідного файлу:	events.out
Обмеження використання часу:	2.5 seconds
Обмеження використання пам'яті:	512 megabytes

Як відомо, мешканці царства Потоколяндія — дуже педантичні люди. І навіть коли справа доходить до свят, вони завжди хочуть бути впевненими в тому, що все пройде дуже добре. Тому розклад всіх свят складений на сто років вперед. Козак Вус вирішив запросити свого друга — Козака Вуха приїхати в одне із міст царства і відвідати як можна більше свят.

У царстві n міст, які сполучені $n - 1$ двонаправленою дорогою так, що з будь-якого міста можна дістатися до іншого, можливо, відвідуючи інші міста. Для того, щоб пройти по i -й дорозі, потрібно l_i днів.

Кожне свято в Потоколяндії характеризується номером міста c_i , в якому воно буде проходити, і номером дня d_i , в який буде відбуватися свято. Козак Вус не витрачає багато часу на святкування. Тому, якщо він святкує в i -ий день, то він може в той самий день виїхати, приїхати в інше місто наступного дня (якщо є така дорога, що $l_i = 1$) та святкувати (якщо таке свято є).

Друг Козака Вуса — такий щасливчик, що день його прибуття до царства має номер 0 в календарі, причому спочатку він може приїхати в будь-яке місто царства. Козак Вус вирішив дізнатися, яку максимальну кількість свят може відвідати його друг. Для цього він звернувся за допомогою до Вас. Допоможіть йому це зробити!

Формат вхідних даних

Перший рядок містить одне ціле число n ($1 \leq n \leq 2 \cdot 10^5$) — кількість міст у царстві.

Кожен з наступних $n - 1$ рядків містить по три цілих числа a_i , b_i та l_i ($1 \leq a_i, b_i \leq n$, $1 \leq l_i \leq 10^9$) — номери міст, які з'єднує дорога, і кількість днів, яка необхідна для її подолання. Гарантується, що граф зв'язний.

Наступний рядок містить одне ціле число m ($1 \leq m \leq 2 \cdot 10^5$) — кількість свят у царстві.

Кожен з наступних m рядків містить по два цілих числа c_i і d_i ($1 \leq c_i \leq n$, $1 \leq d_i \leq 10^9$) — номер міста та номер дня, в який буде відбуватися i -те свято.

Формат вихідних даних

Виведіть одне число — максимальну кількість свят, які зможе відвідати Козак Вус.

Приклади

events.in	events.out
4 1 2 1 2 3 1 2 4 3 4 1 3 2 4 3 1 4 5	3
11 2 1 2 3 2 5 4 1 5 5 2 4 6 5 1 7 1 2 8 3 4 9 6 2 10 7 2 11 2 2 9 1 67 1 34 11 16 5 97 4 70 2 20 2 61 2 26 2 70	8

Примітка

Спочатку Козак Вух може прибути до міста 3 та зачекати один день до святкування. Після цього, в перший день він може за два дні переїхати до міста 1, де у третій день буде свято. Так само у третій день він може виїхати до міста 2, де у четвертий день також буде свято. Але до останнього свята він вже не встигне доїхати, бо потрібно 3 дні, щоб дібратись до міста 4. Таким чином, він відвідав 3 свята.

Оцінювання

№	Обмеження			Бали
	n	m	l_i, d_i	
1	$1 \leq n \leq 100$	$1 \leq m \leq 9$	$1 \leq l_i, d_i \leq 100$	14
2	$1 \leq n \leq 2000$	$1 \leq m \leq 2000$	$1 \leq l_i, d_i \leq 5000$	17
3	$1 \leq n \leq 5000$	$1 \leq m \leq 5000$	$1 \leq l_i, d_i \leq 10^9$	28
4	$1 \leq n \leq 10^5$	$1 \leq m \leq 10^5$		22
5	$1 \leq n \leq 2 \cdot 10^5$	$1 \leq m \leq 2 \cdot 10^5$		19