

Московская олимпиада школьников. Информатика. 9 класс. Дистанционный этап, 2024/25

20 дек 2024 г., 10:00 — 20 янв 2025 г., 23:59

100 баллов

Шутка с таймером

В семье две сестры, Аня и Катя. Однажды, в 7:30 утра, Аня установила таймер на телефоне Кати, который должен сработать через k минут. Катя не подозревала об этом и отправилась в школу с телефоном. Занятия у Кати начинаются в 9:00. Каждый урок длится n минут, а перерыв между ними составляет m минут. Всего у неё запланировано t уроков. Определите, на каком уроке прозвонит таймер и удастся ли Ане осуществить свою шутку. Учтите, что звонок на урок или с урока начинается в первую секунду минуты, а таймер срабатывает в тридцатую секунду минуты.

Вам необходимо выяснить номер урока, когда прозвонит таймер. Если таймер сработает во время перемены, до начала первого урока или после завершения последнего, выведите -1 .

Формат входных данных

Четыре целых числа даны каждый в своей строке в следующем порядке: k, n, m, t — число минут на таймере, длина урока, длина перемены и количество уроков,
 $1 \leq k \leq 1000, 10 \leq n \leq 100, 10 \leq m \leq 100, 2 \leq t \leq 10$.

Формат выходных данных

Выведите единственное число — номер урока в течение которого сработал будильник или -1 , если будильник сработал во время перемены, до первого или после последнего урока, т. е. вне уроков.

Критерии оценивания

Каждый тест оценивается независимо.

Примеры

```
95
45
10
5
```

```
1
```

```
40
45
10
5
```

```
-1
```

```
400
45
15
5
```

```
-1
```

Ограничения

Время выполнения: 1 секунда

Память: 256 МВ

100 баллов

Подходящие числа

Математикам обычно нравятся только особенные числа, например, простые, у которых нет делителей кроме 1 и самого числа, или степени двойки. Назовем такие числа *подходящими*. Поэтому, если к ужину смарт-часы показывают некоторое количество пройденных за день шагов, и оно не является подходящим числом, то математик сделает минимально возможное дополнительное число шагов, чтобы счетчик шагов показывал подходящее число.

Помогите математикам в определении ближайшего к текущим показаниям часов подходящего числа.

Формат входных данных

В первой строке дано одно целое число n ($1 \leq n \leq 10^5$) — число дней, в которые снимались показания часов.

Во второй строке даны n целых чисел a_i ($0 \leq a_i \leq 10^7$) — вечерние показания часов в течение n дней.

Формат выходных данных

Выведите n чисел: для каждого a_i ближайшее подходящее число, не меньшее его.

Критерии оценивания

Подзадача	Баллы	Ограничения
0	0	Тесты из условия
1	25	$n \leq 1000, a_i \leq 1000$
2	17	$n \leq 1000$
3	28	$a_i \leq 1000$
4	30	Без дополнительных ограничений

Примеры

```
5
2020 1023 0 101 10000
2027 1024 1 101 10007
```

Ограничения

Время выполнения: 4 секунды

Память: 256 МВ

100 баллов

Симметричная впадина

Массив натуральных чисел называется *впадиной*, если он сначала строго убывает, затем строго возрастает. Длина впадины как минимум 3. Например, 4 3 2 6 9 — это впадина, потому что первые три числа в нем строго убывают, а потом с третьего по пятое число — строго возрастают.

Массив натуральных чисел называется *палиндромом*, если он один и тот же при записи слева направо и справа налево. Например, 1 3 2 3 1 — это палиндром.

Подмассив массива — массив, полученный при удалении любого префикса (начала) и любого суффикса (конца) исходного массива. Он состоит из подряд идущих элементов исходного массива.

Для заданного массива определите, какова длина самого длинного подмассива данного массива, являющегося одновременно и впадиной, и палиндромом. Например, 4 3 2 3 4 — это палиндром и впадина одновременно.

Формат входных данных

В первой строке дано одно целое число n — размер массива, $1 \leq n \leq 10^6$. Во второй строке n целых чисел через пробел a_i — элементы массива, $1 \leq a_i \leq 10^9$.

Формат выходных данных

Выведите единственное целое число — максимальную длину палиндромной ямы или -1 , если такой нет. Напомним, что длина ямы должна быть как минимум 3.

Критерии оценивания

Все тесты разбиты на группы со следующими ограничениями:

$1 \leq n \leq 50$, 15 баллов

$1 \leq n \leq 300$, 15 баллов

$1 \leq n \leq 7000$, 25 баллов

$1 \leq n \leq 1000000$, 45 баллов

Очередная группа проверяется только после прохождения всех тестов всех предыдущих групп.

Примеры

```
8
11 9 5 2 1 2 5 7
```

```
5
```

Ограничения

Время выполнения: 2 секунды

Память: 256 МВ

100 баллов

Дорога в школу

Будем считать, что дорога в школу начинается в момент времени 0 и состоит из n участков. Ученик передвигается по i -му участку пути со скоростью V_i в течение T_i единиц времени. Требуется q запросов определить, чему равна **средняя скорость** на части пути ученика в различные промежутки времени.

Средняя скорость на части пути вычисляется как отношение всего преодолённого расстояния к затраченному на это времени. $V = \frac{S}{T}$.

Формат входных данных

В первой строке записаны два натуральных числа n, q ($1 \leq n, q \leq 10^5$) — количество участков пути и количество запросов.

В следующих n строках записаны пары целых чисел V_i, T_i ($1 \leq V_i, T_i \leq 10^5$) — информация об участках пути ученика в школу.

В следующих q строках даны запросы. Каждая строка содержит два целых числа

L_q, R_q ($0 \leq L_q < R_q \leq \sum_{i=1}^N T_i$) — начальный и конечный моменты времени интересующего промежутка.

Формат выходных данных

Для каждого запроса выведите среднюю скорость на части пути в соответствующем промежутке времени. Ответ выведите с точностью не менее чем четыре знака после точки (но, например, если ответ целый, лишние нули после точки можно не выводить, и саму точку можно не ставить).

Критерии оценивания

Тесты разделены на несколько подзадач. Баллы выставляются только за подзадачу в целом. Следующая подзадача проверяется только при прохождении всех тестов предыдущих подзадач.

Подзадача 1 (16 баллов): $N \leq 100, Q \leq 100, \sum_{i=1}^N T_i \leq 1000$.

Подзадача 2 (14 баллов): на N и Q нет дополнительных ограничений, $\sum_{i=1}^N T_i \leq 10^5$.

Подзадача 3 (15 баллов): $N \leq 100, Q \leq 100$, на $\sum_{i=1}^N T_i$ нет дополнительных ограничений.

Подзадача 4 (10 баллов): Гарантируется, что для каждого запроса q в момент времени L_q и в момент времени R_q Ученик находился на одном и том же участке пути (участок может быть разным для разных запросов).

Подзадача 5 (15 баллов): Гарантируется, что для каждого запроса q выполняется следующее: либо ученик преодолел участок пути целиком, либо вообще не передвигался по нему.

Подзадача 6 (30 баллов): Ограничения из условия.

Примеры

```
5 7
3 4
1 1
4 2
9 5
2 4
0 16
9 10
5 7
13 15
2 14
0 6
5 10
```

```
4.6250
9.0000
```

4.0000

2.0000

5.3333

2.8333

7.0000

Ограничения

Время выполнения: 2 секунды

Память: 256 MB

100 баллов

Шахматный путь

Друзья играют в следующую игру: один называет шахматную фигуру, второй — стартовую клетку на стандартной шахматной доске, третий — конечную клетку. Далее они соревнуются — кто раньше определит минимальное количество ходов, за которое эта фигура может добраться из **стартовой** клетки в **конечную**. Друзья договорились, что все фигуры будут белыми, а шахматная доска будет считаться свободной от других фигур.

Ряды на доске обозначаются буквами и цифрами, начиная от левого нижнего углового поля $a1$ — чёрного цвета. Вертикальные ряды — латинскими буквами $a-h$, горизонтальные — цифрами 1–8.

Правила передвижения фигур (адаптированы под задачу):

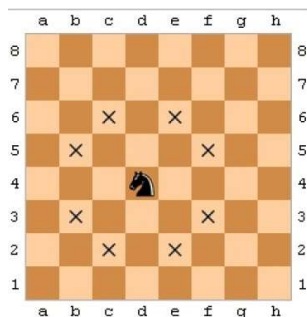
Король (KING) передвигается со своего поля на одно из свободных смежных полей.

Ферзь (QUEEN) ходит по вертикалям, диагоналям и горизонталям, на которых он находится.

Ладья (ROOK) ходит по вертикалям и горизонталям, на которых она находится.

Слон (BISHOP) ходит по диагоналям, на которых он находится.

Конь (KNIGHT) может пойти на одно из полей, ближайших к тому, на котором он стоит, но не на той же самой горизонтали, вертикали или диагонали.



Пешка (PAWN) передвигается на одно поле только вперёд. Если стартовая клетка соответствует стартовой позиции белой пешки в настоящей игре (это горизонталь $a2-h2$), пешка может пойти как на одну, так и на две клетки вперёд в первом ходе. Любая пешка, достигающая крайней горизонтали (для белых это горизонталь $a8-h8$), должна быть тем же ходом заменена на ферзя того же цвета, что и пешка. Если в тестовом случае пешка изначально находится на горизонтали $a8-h8$, то она сразу заменяется на ферзя (эта замена ходом не считается).

Формат входных данных

В первой строке записано число T ($1 \leq T \leq 24576$) — количество случаев, на которые нужно получить ответ.

В следующих T строках описаны сами случаи. Каждое описание начинается со строки, задающей фигуру:

KING — король, QUEEN — ферзь, ROOK — ладья, BISHOP — слон, KNIGHT — конь, PAWN — пешка.

Далее, через пробел, записаны координаты стартовой клетки в формате cx , где c — строчная английская буква от 'a' до 'h', обозначающая ряд, а x — натуральное число от 1 до 8, обозначающее столбец. Далее, через пробел, задана конечная клетка в аналогичном формате.

Формат выходных данных

Для каждого тестового случая выведите ответ в отдельной строке — минимальное количество ходов, которое необходимо сделать указанной фигуре, чтобы прийти из стартовой клетки в конечную клетку.

В случае, если конечная клетка недостижима из стартовой, выведите -1 .

Замечание

В примере у первой пешки стартовая позиция совпадает со стартовой позицией в настоящей игре, поэтому она может достичь конечной за один ход. Для второй пешки это не так, поэтому она может ходить вперед только в соседнюю клетку.

Критерии оценивания

Все тесты оцениваются независимо.

Во многих тестах во всех случаях упоминается только одна из фигур.

Примеры

```
9
ROOK c5 h2
KING a1 a2
PAWN b2 b4
PAWN b1 b4
BISHOP e5 e6
PAWN c5 e5
QUEEN e5 g7
KNIGHT a1 c1
KNIGHT a1 a1
```

```
2
1
1
3
-1
5
1
2
0
```

Ограничения

Время выполнения: 1 секунда

Память: 256 MB

100 баллов

Правильные подматрицы

Дана матрица (таблица) из n строк и m столбцов, заполненная строчными буквами латинского алфавита.

Назовем матрицу *правильной*, если в ней встречаются **ровно две различные** буквы, и они расположены в шахматном порядке (одна буква на местах белых клеток, вторая — чёрных).

Следующие матрицы **являются** правильными:

<i>o</i>	<i>x</i>	<i>o</i>
<i>x</i>	<i>o</i>	<i>x</i>
<i>o</i>	<i>x</i>	<i>o</i>

<i>x</i>	<i>o</i>	<i>x</i>
<i>o</i>	<i>x</i>	<i>o</i>
<i>x</i>	<i>o</i>	<i>x</i>

<i>a</i>
<i>b</i>
<i>a</i>

Следующие матрицы **не являются** правильными:

<i>o</i>	<i>x</i>	<i>o</i>
<i>x</i>	<i>x</i>	<i>x</i>
<i>o</i>	<i>x</i>	<i>o</i>

<i>a</i>	<i>a</i>	<i>a</i>
<i>a</i>	<i>a</i>	<i>a</i>
<i>a</i>	<i>a</i>	<i>a</i>

<i>b</i>

Требуется найти количество *правильных* подматриц данной матрицы.

Подматрицей называется любая прямоугольная часть исходной матрицы. Она получается из исходной матрицы сначала отбрасыванием нескольких (возможно, 0) подряд идущих строк в начале и нескольких строк (возможно, 0) в конце, а затем в полученной матрице можно отбросить несколько (возможно, 0) столбцов в начале и несколько (возможно, 0) столбцов в конце.

Формат входных данных

В первой строке даны два целых числа n и m ($2 \leq n, m \leq 300$) — количество строк и столбцов в матрице соответственно.

В каждой из следующих n строк задана последовательность, состоящая из m строчных букв латинского алфавита.

Формат выходных данных

Выведите единственное число — количество правильных подматриц данной матрицы.

Критерии оценивания

Все тесты оцениваются независимо.

Примеры

2 2
aa
aa

0

2 2
ab
cd

4

2 2
ab
ba

5

Ограничения

Время выполнения: 3 секунды

Память: 256 МВ

100 баллов

Доминирующий элемент

Дан массив из n натуральных чисел, проиндексированный с 1 до n включительно. Рассмотрим некоторый отрезок массива $[l, r]$ включительно, и назовем элемент x *доминирующим* на этом отрезке, если он встречается на нём строго больше раз, чем все остальные элементы вместе взятые.

Дано q запросов, в каждом запросе даны $[l_q, r_q]$ — границы отрезка. Найдите доминирующий элемент этого отрезка либо сообщите, что на этом отрезке его нет.

Формат входных данных

В первой строке вам даны два числа n, q ($1 \leq n, q \leq 10^5$) — количество элементов в массиве и количество запросов.

Во второй строке задан массив из n натуральных чисел a_i ($1 \leq a_i \leq 10^9$).

В следующих q строках заданы запросы, в каждой новой строке по два числа l_q, r_q ($1 \leq l_q \leq r_q \leq n$) — границы отрезка в очередном запросе.

Формат входных данных

Для каждого запроса в новой строке выведите значение доминирующего элемента на этом отрезке, либо -1 , если доминирующего элемента на отрезке нет.

Критерии оценивания

Подзадача 1 (18 баллов): $n \leq 500, q \leq 500$.

Подзадача 2 (14 баллов): $n \leq 3000, q \leq 3000$.

Подзадача 3 (20 баллов): $n \leq 20000, q \leq 20000$.

Подзадача 4 (16 баллов): В массиве присутствует не более 10 различных значений.

Подзадача 5 (32 балла): Ограничения из условия.

Баллы за подзадачу выставляются только при прохождении всех тестов подзадачи. Подзадача проверяется, если все тесты всех предыдущих подзадач пройдены.

Примеры

```
5 4
3 2 3 1 3
1 3
3 5
1 5
1 4
```

```
3
3
3
-1
```

Ограничения

Время выполнения: 4 секунды

Память: 512 МВ

100 баллов

Граф. Просто граф

Дан неориентированный связный граф, состоящий из n вершин, пронумерованных от 1 до n , и m ребер, а также q запросов. В каждом из запросов, вам требуется по заданным различным вершинам u и v , найти количество различных простых путей, соединяющих u и v . Напомним, что путь называется простым, если этот путь проходит по каждой из вершин графа не более чем по одному разу.

Формат входных данных

В первой строке записаны целые неотрицательные числа n и m ($1 \leq n \leq 200\,000$, $0 \leq m \leq n + 4$) — количество вершин и ребер графа.

Далее в следующих m строках записано по паре целых чисел u, v , $1 \leq u, v \leq n$, $u \neq v$.

В следующей строке записано целое число q , $1 \leq q \leq 2\,000\,000$ количество запросов. В следующих q строках записано по паре целых чисел u, v , $1 \leq u, v \leq n$, $u \neq v$, описывающих запрос.

Гарантируется, что в графе не существует петель и кратных ребер.

Формат выходных данных

Для каждого запроса выведите в отдельной строке количество различных путей между вершинами u и v .

Критерии оценивания

Каждый тест оценивается независимо. Как минимум в 60% тестов n и m не превосходят 50.

Примеры

```
3 3
1 2
2 3
3 1
3
1 2
3 1
2 3
```

```
2
2
2
```

```
4 4
1 2
2 3
3 1
1 4
6
1 2
1 3
1 4
2 3
2 4
3 4
```

```
2
2
1
2
2
2
```

Ограничения

Время выполнения: 4 секунды

Память: 512 МВ