

FAKEBS: Fake Binary Search

Легенда, описанная далее, переформулирована и упрощена переводчиком, чтобы читатель мог лучше понять условие задачи. Оригинальную легенду вы можете прочитать на странице задачи в контексте.

Условие:

В классе Шефа учатся Q студентов. Учитель дал всем простое задание: “Напишите функцию, которая принимает в качестве аргумента массив A , содержащий уникальные элементы и число X , и возвращает номер позиции элемента X в массиве A .”

Учитель ожидает увидеть алгоритм линейного поиска, но так как Шеф — отличный программист, он написал функцию бинарного поиска:

```
integer binary_search(array a, integer n, integer x):
    integer low, high, mid
    low := 1
    high := n
    while low ≤ high:
        mid := (low + high) / 2
        if a[mid] == x:
            break
        else if a[mid] is less than x:
            low := mid+1
        else:
            high := mid-1
    return mid
```

Все одноклассники Шефа скопировали его код для решения задачи.

Но только потом Шеф понял, что забыл отсортировать массив, и поэтому бинарный поиск может не работать. К радости Шефа, учитель сегодня устал и попросил его проверить работы других учеников. Код каждого ученика оценивается по входному массиву A и целому числу X , и проверяется, правильно ли был возвращен индекс. Учитель достаточно ленив, поэтому всем ученикам он предоставляет один и тот же массив A , но различные значения X .

Шеф должен ввести входные данные для каждой программы. Он решает, что при вводе входного массива для каждого решения он не будет использовать входной массив, который задан изначально, а массив, созданный путем замены некоторых пар элементов этого исходного входного массива. Однако он не может изменить положение элемента X так как это было бы подозрительно.

Ваша задача — для каждого из Q студентов найти минимальное число обменов, которое потребуется для того, чтобы решение выводило правильный ответ.

Формат ввода:

Первая строка содержит единственное целое число T — число тестовых случаев.

Далее следует описание тестов в следующем формате:

Первая строка каждого теста содержит разделенные пробелами целые числа N и Q — число элементов в последовательности и число студентов, соответственно.

Вторая строка каждого теста содержит N разделенных пробелами целых чисел A_1, A_2, \dots, A_N .

Каждая из следующих Q строк каждого теста содержит описание запросов, каждая из которых содержит единственное целое число X .

Формат вывода:

Для каждого тестового случая выведите единственное целое число — минимальное число обменов, которое требуется для того, чтобы алгоритм выводил правильный ответ. Если же это невозможно то сделать, то выведите -1.

Ограничения:

- $1 \leq T \leq 10$
- $1 \leq N, Q \leq 10^5$
- $1 \leq A_i \leq 10^9$ для каждого i
- $1 \leq X \leq 10^9$
- все элементы массива A попарно различны
- для каждого запроса элемент X принадлежит A
- сумма всех N во всех тестовых случаях $\leq 5 * 10^5$
- сумма всех Q во всех тестовых случаях $\leq 5 * 10^5$

Подзадачи:

- Подзадача 1 (20 баллов): $1 \leq N \leq 10$
- Подзадача 2 (30 баллов): $1 \leq A_i \leq 10^6, 1 \leq X \leq 10^6$
- Подзадача 3 (50 баллов): ограничения из условия

Примеры тестов:

Входные данные:

```
1
7 7
3 1 6 7 2 5 4
1
2
3
4
5
6
7
```

Выходные данные:

```
0
1
1
2
1
0
0
```

Пояснения:

Тест 1:

- Запрос 1: Алгоритм работает корректно без перестановок
- Запрос 2: Возможное решение — поменять местами A_2 и A_4 .
- Запрос 3: Возможное решение — поменять местами A_2 и A_6 .
- Запрос 4: Возможное решение — поменять местами A_2 и A_4 и A_5 и A_6 .
- Запрос 5: Возможное решение — поменять местами A_2 и A_4 .
- Запрос 6: Алгоритм работает корректно без перестановок
- Запрос 7: Алгоритм работает корректно без перестановок