

Тренировки по алгоритмам 5.0 от Яндекса — Занятие 4 (Бинарный поиск)

С. Саруман

Ограничение времени	2 секунды
Ограничение памяти	256Mb
Ввод	стандартный ввод или input.txt
Вывод	стандартный вывод или output.txt

Как известно, Саруман Радужный очень любит порядок. Поэтому все полки его войска стоят друг за другом, причем каждый следующий полк содержит количество орков не меньше, чем предыдущий.

Перед тем как напасть на Хельмову Падь, Саруман решил провести несколько вылазок для разведки. Чтобы его отряды никто не заметил, он решил каждый раз отправлять несколько подряд идущих полков так, чтобы суммарное количество орков в них было равно определенному числу. Так как это всего лишь разведка, каждый полк после вылазки возвращается на свое место. Задачу выбрать нужные полки он поручил Гриму Змеиному Языку. А Грима не поскупится на вознаграждение, если вы ему поможете.

Формат ввода

В первой строке входного файла находится два целых числа: n ($1 \leq n \leq 2 \cdot 10^5$) — количество полков и m ($1 \leq m \leq 2 \cdot 10^5$) — количество предстоящих вылазок.

В следующей строке записано n чисел a_i , где a_i — число орков в i -ом полке ($1 \leq a_i \leq 10^9, a_i \leq a_{i+1}$).

Далее в m строках записаны запросы вида: количество полков l ($1 \leq l \leq n$), которые должны будут отправиться в эту вылазку, и суммарное количество орков в этих полках s ($1 \leq s \leq 2 \cdot 10^{16}$).

Формат вывода

Для каждого запроса выведите номер полка, с которого начнутся те l , которые необходимо отправить на вылазку. Если таких полков несколько, выведите любой. Если же так выбрать полки нельзя, выведите -1 .

Пример

Ввод <input type="text"/>	Вывод <input type="text"/>
5 2	1
1 3 5 7 9	2
2 4	
1 3	

```

1 def binSearchLeft(lf, rg, check, checkparams):
2     """
3     Бинарный поиск самого левого вхождения числа
4     :param lf: левая граница
5     :param rg: правая граница
6     :param check: функция проверки
7     :param checkparams: параметры функции проверки
8     :return: значение, на котором сошелся бинарный поиск
9     """
10    while lf < rg:
11        mid = (lf + rg) // 2
12        if check(mid, checkparams):
13            rg = mid
14        else:
15            lf = mid + 1
16    return lf
17
18 def binSearchRight(lf, rg, check, checkparams):
19    """
20    Бинарный поиск самого правого вхождения числа
21    :param lf: левая граница
22    :param rg: правая граница
23    :param check: функция проверки
24    :param checkparams: параметры функции проверки
25    :return: значение, на котором сошелся бинарный поиск
26    """
27    while lf < rg:
28        mid = (lf + rg + 1) // 2
29        if check(mid, checkparams):
30            lf = mid
31        else:
32            rg = mid - 1
33    return lf
34
35 def check_target_left(m, checkparams): # проверка для левого бин поиска
36     count_orcs, sum_orcs = checkparams
37     #print(m, prof_sum[m] + count_orcs - 1, prof_sum[m - 1], sum_orcs)
38

```

Отправить

Предыдущая

Следующая