

Тренировки по алгоритмам 5.0 от Яндекса — Занятие 2 (Линейный поиск)

🕒 2 апр 2024, 06:51:37
старт: 6 мар 2024, 22:30:00
финиш: 20 мар 2024, 20:00:00
длительность: 13д. 21ч.
...

Объявления жюри

📌 Ваше участие в соревновании завершено. Вы можете дорешивать задачи и отправлять решения вне соревнования

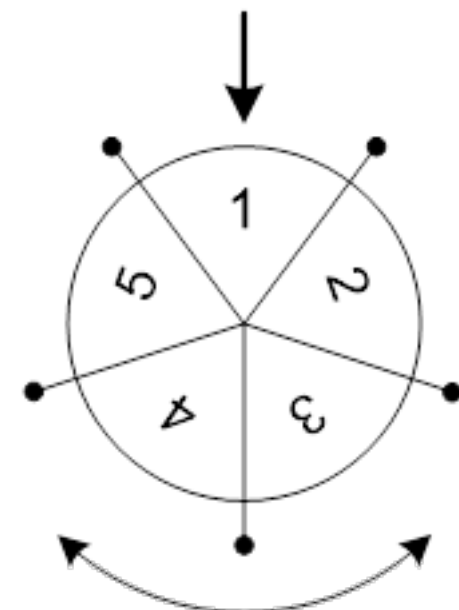
Положение участников Задачи Посылки

F. Колесо Фортуны

Ограничение времени	1 секунда
Ограничение памяти	64Mb
Ввод	стандартный ввод или input.txt
Вывод	стандартный вывод или output.txt

Развлекательный телеканал транслирует шоу «Колесо Фортуны». В процессе игры участники шоу крутят большое колесо, разделенное на сектора. В каждом секторе этого колеса записано число. После того как колесо останавливается, специальная стрелка указывает на один из секторов. Число в этом секторе определяет выигрыш игрока.

Юный участник шоу заметил, что колесо в процессе вращения замедляется из-за того, что стрелка задевает за выступы на колесе, находящиеся между секторами. Если колесо вращается с угловой скоростью v градусов в секунду, и стрелка, переходя из сектора X к следующему сектору, задевает за очередной выступ, то текущая угловая скорость движения колеса уменьшается на k градусов в секунду. При этом если $v \leq k$, то колесо не может преодолеть препятствие и останавливается. Стрелка в этом случае будет указывать на сектор X .



Юный участник шоу собирается вращать колесо. Зная порядок секторов на колесе, он хочет заставить колесо вращаться с такой начальной скоростью, чтобы после остановки колеса стрелка указала на как можно большее число. Колесо можно вращать в любом направлении и придавать ему начальную угловую скорость от a до b градусов в секунду.

Требуется написать программу, которая по заданному расположению чисел в секторах, минимальной и максимальной начальной угловой скорости вращения колеса и величине замедления колеса при переходе через границу секторов вычисляет максимальный выигрыш.

Формат ввода

Первая строка входного файла содержит целое число n — количество секторов колеса ($3 \leq n \leq 100$).

Вторая строка входного файла содержит n положительных целых чисел, каждое из которых не превышает 1000 — числа, записанные в секторах колеса. Числа приведены в порядке следования секторов по часовой стрелке. Изначально стрелка указывает на первое число.

Третья строка содержит три целых числа: a , b и k ($1 \leq a \leq b \leq 10^9$, $1 \leq k \leq 10^9$).

Формат вывода

В выходном файле должно содержаться одно целое число — максимальный выигрыш.

Пример 1

Ввод	Вывод
5 1 2 3 4 5 3 5 2	5

Пример 2

Ввод	Вывод
5 1 2 3 4 5 15 15 2	4

Пример 3

Ввод	Вывод
5 5 4 3 2 1 2 5 2	5

Примечания

В первом примере возможны следующие варианты: можно придать начальную скорость колесу равную 3 или 4, что приведет к тому, что стрелка преодолеет одну границу между секторами, или придать начальную скорость равную 5, что позволит стрелке преодолеть 2 границы между секторами. В первом варианте, если закрутить колесо в одну сторону, то выигрыш получится равным 2, а если закрутить его в противоположную сторону, то — 5. Во втором варианте, если закрутить колесо в одну сторону, то выигрыш будет равным 3, а если в другую сторону, то — 4.

Во втором примере возможна только одна начальная скорость вращения колеса — 15 градусов в секунду. В этом случае при вращении колеса стрелка преодолеет семь границ между секторами. Тогда если его закрутить в одном направлении, то выигрыш составит 4, а если в противоположном направлении, то — 3.

Наконец, в третьем примере оптимальная начальная скорость вращения колеса равна 2 градусам в секунду. В этом случае стрелка вообще не сможет преодолеть границу между секторами, и выигрыш будет равен 5.

Язык Python 3.9 (PyPy 7.3.11) ▾

Набрать здесь Отправить файл

```
1 def rotation(array, min_sector, max_sector):
2     N = len(array)
3     profit = float('-inf')
4
5     cur_step, steps = 0, (max_sector-min_sector+1) % N
6     cur_sector = min_sector
7     while cur_step < steps:
8         profit = max(profit, array[cur_sector])
9         cur_sector = (cur_sector+1) % N
10        cur_step += 1
11
12    return profit
13
14 def answer(numbers, A, B, K):
15     N = len(numbers)
16
17     min_steps = (A-1) // K
18     max_steps = (B-1) // K
19
20     total_sectors = max_steps-min_steps+1
21     if total_sectors >= N: return max(numbers)
22
23     min_sector = min_steps % N
24     max_sector = max_steps % N
25
26     max_profit1 = rotation(numbers, min_sector, max_sector)
27     max_profit2 = rotation(numbers, (-max_sector) % N, (-min_sector) % N)
28
29     return max(max_profit1, max_profit2)
30
31 def main():
32     with open('input.txt') as fin:
33         N = int(fin.readline())
34         numbers = [int(x) for x in fin.readline().split()]
35         A, B, K = [int(x) for x in fin.readline().split()]
36
37         print(answer(numbers, A, B, K))
38
```

Отправить 📌 осталось 100 попыток

Предыдущая Следующая

Время посылки	ID	Задача	Компилятор	Вердикт	Тип посылки	Время	Память	Тест	Баллы
9 мар 2024, 00:38:09	109149125	F	Python 3.9 (PyPy 7.3.11)	OK	-	201ms	28.32Mb	-	-
9 мар 2024, 00:12:42	109147930	F	Python 3.9 (PyPy 7.3.11)	WA	-	208ms	28.09Mb	11	-
9 мар 2024, 00:02:04	109147438	F	Python 3.9 (PyPy 7.3.11)	WA	-	187ms	28.09Mb	5	-