

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

## Е. Амбициозная улитка

Ограничение времени	5 секунд
Ограничение памяти	256Mb
Ввод	стандартный ввод или input.txt
Вывод	стандартный вывод или output.txt

Домашний питомец мальчика Васи — улитка Петя. Петя обитает на бесконечном в обе стороны вертикальном столбе, который для удобства можно представить как числовую прямую. Изначально Петя находится в точке 0. Вася кормит Петю ягодами. У него есть  $n$  ягод, каждая в единственном экземпляре. Вася знает, что если утром он даст Пете ягоду с номером  $i$ , то поев и набравшись сил, за остаток дня Петя поднимется на  $a_i$  единиц вверх по столбу, но при этом за ночь, потяжелев, съедет на  $b_i$  единиц вниз. Параметры различных ягод могут совпадать.

Пете стало интересно, а как оно там, наверху, и Вася взялся ему в этом помочь. Ближайшие  $n$  дней он будет кормить Петю ягодами из своего запаса таким образом, чтобы максимальная высота, на которой побывал Петя за эти  $n$  дней была максимальной. К сожалению, Вася не умеет программировать, поэтому он попросил вас о помощи. Найдите, максимальную высоту, на которой Петя сможет побывать за эти  $n$  дней и в каком порядке Вася должен давать Пете ягоды, чтобы Петя смог её достичь!

### Формат ввода

В первой строке входных данных дано число  $n$  ( $1 \leq n \leq 5 \cdot 10^5$ ) — количество ягод у Васи. В последующих  $n$  строках описываются параметры каждой ягоды. В  $i + 1$  строке дано два числа  $a_i$  и  $b_i$  ( $0 \leq a_i, b_i \leq 10^9$ ) — то, насколько поднимется улитка за день после того, как съест  $i$  ягоду и насколько опуститься за ночь.

### Формат вывода

В первой строке выходных данных выведите единственное число — максимальную высоту, которую сможет достичь Петя, если Вася будет его кормить оптимальным образом. В следующей строке выведите  $n$  различных целых чисел от 1 до  $n$  — порядок, в котором Вася должен кормить Петю ( $i$  число в строке соответствует номеру ягоды, которую Вася должен дать Пете в  $i$  день чтобы Петя смог достичь максимальной высоты).

### Пример 1

Ввод	<input type="text"/>	Вывод	<input type="text"/>
3		10	
1 5		2 3 1	
8 2			
4 4			

### Пример 2

**Ввод** **Вывод** 2  
7 6  
7 410  
2 1

## Примечания

Во втором примере изначально улитка находится на высоте 0. Пусть сначала Петя накормит её второй ягодой, а затем первой. После того как она съест вторую ягоду, за день она поднимется на 7 (и окажется на высоте 7), а за ночь опустится на 4 (и окажется на высоте 3). После того как она съест первую ягоду, за день она поднимется на 7 (и окажется на высоте 10), а за ночь опустится на 6 (и окажется на высоте 4).

Таким образом, максимальная высота, на которой побывает улитка при данном порядке кормления, равна 10. Нетрудно видеть, что если Петя накормит улитку сначала первой ягодой, а затем второй, то максимальная высота, на которой побывает улитка, будет меньше.

Язык Python 3.12.1

Набрать здесь

Отправить файл

```
1 # считываем данные
2 n = int(input().strip()) # количество ягод
3 answer = [0] * n # массив ответа
4 berries = [''] * (n + 1) # массив значений ягод (поднимется за день, разница движения)
5
6 pos_diff_sum = 0 # кумулятивная сумма положительных разниц движений
7 for i in range(1, n + 1): # заполняем berries
8     a, b = map(int, input().split())
9     diff = a - b
10    berries[i] = (a, diff)
11    pos_diff_sum += diff if diff > 0 else 0
12
13 # находим ягоду "достижения" макс высоты
14 idx_max = - 1
15 max_h = float('-inf')
16 for i in range(1, n + 1):
17     if berries[i][1] <= 0:
18         if pos_diff_sum + berries[i][0] > max_h:
19             idx_max = i
20             max_h = pos_diff_sum + berries[i][0]
21     else:
22         if pos_diff_sum - berries[i][1] + berries[i][0] > max_h:
23             idx_max = i
24             max_h = pos_diff_sum - berries[i][1] + berries[i][0]
25
26 # скармливаем улитке ягоды с положительной разницей
27 idx_answer = 0 # индекс в ответе
28 for i in range(1, n + 1):
29     if berries[i][1] > 0 and i != idx_max:
30         answer[idx_answer] = i
31         idx_answer += 1
32 # скармливаем улитке последнюю ягоду
33 answer[idx_answer] = idx_max
34 idx_answer += 1
35 # скармливаем улитке оставшиеся ягоды
36 for i in range(1, n + 1): # скармливаем улитке ягоды с положительной разницей
37     if berries[i][1] <= 0 and i != idx_max:
38         answer[idx_answer] = i
```

Отправить

Предыдущая

Следующая