

# Яндекс. Тренировки по алгоритмам 2.0, занятие 6 (А)

## Е. Выборы

Ограничение времени	2 секунды
Ограничение памяти	64Mb
Ввод	стандартный ввод или input.txt
Вывод	стандартный вывод или output.txt

В одной демократической стране приближаются парламентские выборы. Выборы проходят по следующей схеме: каждый житель страны, достигший восемнадцатилетнего возраста, отдает свой голос за одну из политических партий. После этого партия, которая набрала максимальное количество голосов, считается победившей на выборах и формирует правительство. Если несколько партий набрали одинаковое максимальное количество голосов, то они должны сформировать коалиционное правительство, что обычно приводит к длительным переговорам.

Один бизнесмен решил выгодно вложить свои средства и собрался поддержать на выборах некоторые партии. В результате поддержки он планирует добиться победы одной из этих партий, которая затем сформирует правительство, которое будет действовать в его интересах. При этом возможность формирования коалиционного правительства его не устраивает, поэтому он планирует добиться строгой победы одной из партий.

Чтобы повлиять на исход выборов, бизнесмен собирается выделить деньги на агитационную работу среди жителей страны. Исследование рынка показало, что для того, чтобы один житель сменил свои политические воззрения, требуется потратить одну условную единицу. Кроме того, чтобы  $i$ -я партия в случае победы сформировала правительство, которое будет действовать в интересах бизнесмена, необходимо дать лидеру этой партии взятку в размере  $p_i$  условных единиц. При этом некоторые партии оказались идеологически устойчивыми и не согласны на сотрудничество с бизнесменом ни за какие деньги.

По результатам последних опросов известно, сколько граждан планируют проголосовать за каждую партию перед началом агитационной компании. Помогите бизнесмену выбрать, какую партию следует подкупить, и какое количество граждан придется убедить сменить свои политические воззрения, чтобы выбранная партия победила, учитывая, что бизнесмен хочет потратить на всю операцию минимальное количество денег.

### Формат ввода

В первой строке вводится целое число  $n$  – количество партий ( $1 \leq n \leq 10^5$ ). Следующие  $n$  строк описывают партии. Каждая из этих строк содержит по два целых числа:  $v_i$  – количество жителей, которые собираются проголосовать за эту партию перед началом агитационной компании, и  $p_i$  – взятка, которую необходимо дать лидеру партии для того, чтобы сформированное ей в случае победы правительство действовало в интересах бизнесмена ( $1 \leq v_i \leq 10^6$ ,  $1 \leq p_i \leq 10^6$  или  $p_i = -1$ ). Если партия является идеологически устойчивой, то  $p_i$  равно  $-1$ . Гарантируется, что хотя бы одно  $p_i$  не равно  $-1$ .

### Формат вывода

В первой строке выведите минимальную сумму, которую придется потратить бизнесмену. Во второй строке выведите номер партии, лидеру которой следует дать взятку. В третьей строке выведите  $n$  целых чисел – количество голосов, которые будут отданы за каждую из партий после осуществления операции. Если оптимальных решений несколько, выведите любое.

### Пример

Ввод	<input type="text"/>	Вывод	<input type="text"/>
3		6	
7 -1		3	

**Ввод** **Вывод** 

2 8

3 2 5

1 2

Язык Python 3.9 (PyPy 7.3.11)

Набрать здесь

Отправить файл

```
1 def get_count_votes(cur_idx, voters, suffix_sum, level):
2     """
3     Функция подсчета получаемых голосов при срезе
4     :param cur_idx: партия для которой считаем получаемые голоса
5     :param voters: массив голосующих
6     :param suffix_sum: массив суффиксных сумм
7     :param level: уровень среза голосов
8     :return: получаемые при срезе голоса
9     """
10    lf, rg = 0, len(voters) - 1 # границы бин поиска
11    # находим партию самую близкую к level, для подсчета голосов от нее и вправо
12    while lf < rg:
13        mid = (lf + rg) // 2 # индекс текущей партии для бин поиска
14        if voters[mid][0] < level:
15            lf = mid + 1
16        else:
17            rg = mid
18    if voters[lf][0] < level: # если level слишком большой - возвращаем 0
19        return 0
20    count_votes = suffix_sum[lf] - level * (len(voters) - lf) # подсчитываем получаемые голоса исходя из партии lf
21    if voters[cur_idx][0] >= level: # если партия для которой считаем >= level, убираем срезанные с нее голоса
22        count_votes -= (voters[cur_idx][0] - level)
23    # возвращаем получаемые при срезе голоса
24    return count_votes
25
26
27 def cost_calculation(voters, cur_idx, suffix_sum):
28     """
29     Функция моделирования перераспределения голосов
30     :param voters: отсортированный массив голосующих избирателей
31     :param cur_idx: проверяемая партия в массиве voters
32     :param suffix_sum: массив суффиксных сумм
33     :return: количество/стоимость полученных голосов, уровень среза голосов, количество возвращаемых голосов
34     """
35    lf, rg = 0, 10 ** 6 # границы бин поиска уровня среза
36    while lf < rg:
37        level = (lf + rg + 1) // 2 # уровень среза голосов
38        count_votes, level_votes, return_votes = get_count_votes(cur_idx, voters, suffix_sum, level)
```

Отправить

Предыдущая