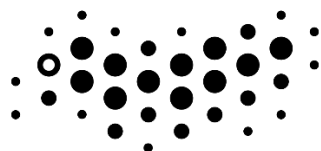


Санкт-Петербургский национальный  
исследовательский университет ИТМО

Факультет систем управления и робототехники



**ITMO UNIVERSITY**

# Алгоритмы и структуры данных. Отчет №4.

Выполнил студент гр. R32362

Лаптев Максим Сергеевич

Санкт-Петербург 2023

## **Содержание**

1. Цель
2. Задача №1604 – В стране дураков
3. Задача №1726 – Кто ходит в гости
4. Задача №1521 – Военные учения
5. Вывод

## **Цель**

Решить данные задачи, написав код, работа которого удовлетворяет условиям

# №1604 «В стране дураков»

Ограничение времени: 1.0 секунды

Ограничение памяти: 64 МБ

Главный бульдог-полицейский Страны Дураков решил ввести ограничение скоростного режима на автомобильной трассе, ведущей от Поля Чудес к пруду Черепахи Тортиллы. Для этого он заказал у Папы Карло  $n$  знаков ограничения скорости. Папа Карло слабо разбирался в дорожном движении и поэтому изготовил знаки с разными ограничениями на скорость: 49 км/ч, 34 км/ч, 42 км/ч, и т.д. Всего получилось  $k$  различных ограничений:  $n_1$  знаков с одним ограничением,  $n_2$  знаков со вторым ограничением, и т.д. ( $n_1 + \dots + n_k = n$ )



Бульдог-полицейский ничуть не расстроился, получив такие знаки, напротив, он решил извлечь из этого экономическую выгоду. Дело в том, что по Правилам дорожного движения Страны Дураков ограничение на скорость действует вплоть до следующего знака. Если на знаке написано число 60, это означает, что участок от данного знака до следующего нужно проехать ровно со скоростью 60 километров в час — не больше и не меньше. Бульдог распорядился расставить знаки так, чтобы обогатившимся на Поле Чудес автолюбителям во время своего движения по трассе приходилось как можно больше раз менять скорость. Для этого нужно расставить имеющиеся знаки в правильном порядке. Если Вы поможете бульдогу это сделать, то он готов будет поделиться с Вами частью своих доходов.

## Исходные данные

В первой строке дано число  $k$  — количество различных типов знаков с ограничением скорости ( $1 \leq k \leq 10000$ ). Во второй строке через пробел перечислены целые положительные числа  $n_1, \dots, n_k$ . Сумма всех  $n_i$  не превосходит 10000.

## Результат

Выведите  $n$  целых чисел в пределах от 1 до  $k$  — порядок, в котором нужно расставить по трассе имеющиеся знаки. Вне зависимости от того, какой знак стоит первым, считается, что, проезжая его, водитель меняет скорость, так как до этого ограничения не действовали. Если задача имеет несколько решений, выведите любое.

## Пример

исходные данные	результат
2 2 2	1 2 1 2

## Исходный код:

```
k = int(input())
x = [[str(index + 1), int(element)] for index, element in
     enumerate(input().split())]
x = (sorted(x, key=lambda item: item[1], reverse=True))
while len(x) > 1:
    prev = 0
    i = 1
    while i < len(x):
        if x[i][1] >= prev:
            if x[0][1] >= x[i][1]:
                print(x[0][0] + " " + x[i][0] + " ", end="")
                x[i][1] -= 1
                x[0][1] -= 1
                prev = x[i][1]
            if x[i][1] == 0: x.pop(i)
            if x[0][1] == 0: x.pop(0)
        i += 1
```

```

        else:
            print(x[i][0] + " " + x[0][0] + " ", end="")
            x[i][1] -= 1
            x[0][1] -= 1
            prev = x[i][1]
            if x[i][1] == 0: x.pop(i)
            if x[0][1] == 0: x.pop(0)

        i += 1
    x = (sorted(x, key=lambda item: item[1], reverse=True))
if len(x) == 1:
    print((x[0][0] + " ") * x[0][1], end = "")

```

В задаче необходимо было как можно больше чередовать дорожные знаки. Для этого я отсортировал количества повторений, сохраняя при этом индексы в двумерном массиве. В цикле же я рассматривал пары значений с самого начала и если у них еще есть повторения, то выводил их. Те знаки, которых больше не осталось, удалялись из массива.

Язык программирования: python

## Результат

ID	Дата	Автор	Задача	Язык	Результат проверки	№ теста	Время работы	Выделено памяти
<a href="#">10218336</a>	00:57:43 27 мар 2023	<a href="#">Maxim</a>	<a href="#">1604. В Стране Дураков</a>	PyPy 3.8 x64	Accepted		0.234	8 804 КБ

# №1726 «Кто ходит в гости»

Ограничение времени: 1.0 секунды

Ограничение памяти: 64 МБ

Программный комитет школьных соревнований по программированию, проходящих в УрГУ — многочисленная, весёлая и дружная команда. Дружная настолько, что общения в университете им явно не хватает, поэтому они часто ходят друг к другу в гости. Все ребята в программном комитете очень спортивные и ходят только пешком.

Однажды хранитель традиций олимпиадного движения УрГУ подумал, что на пешне прогулки от дома к дому члены программного комитета тратят слишком много времени, которое могли бы вместо этого потратить на придумывание и подготовку задач. Чтобы доказать это, он решил посчитать, какое расстояние в среднем преодолевают члены комитета, когда ходят друг к другу в гости. Хранитель традиций достал карту Екатеринбурга, нашёл на ней дома всех членов программного комитета и выписал их координаты. Но координат оказалось так много, что хранитель не смог справиться с этой задачей самостоятельно и попросил вас помочь ему.

Город Екатеринбург представляет собой прямоугольник со сторонами, ориентированными по сторонам света. Все улицы города идут строго с запада на восток или с севера на юг, проходя через весь город от края до края. Дома всех членов программного комитета расположены строго на пересечении каких-то двух перпендикулярных улиц. Известно, что все члены комитета ходят только по улицам, поскольку идти по тротуару гораздо приятнее, чем по дворовым тропинкам. И, конечно, при переходе от дома к дому они всегда выбирают кратчайший путь. Программный комитет очень дружный, и все его члены ходят в гости ко всем одинаково часто.

## Исходные данные

Первая строка содержит целое число  $n$  — количество членов программного комитета ( $2 \leq n \leq 10^5$ ). В  $i$ -й из следующих  $n$  строк через пробел записаны целые числа  $x_i, y_i$  — координаты дома  $i$ -го члена программного комитета ( $1 \leq x_i, y_i \leq 10^6$ ).

## Результат

Выведите среднее расстояние, которое проходит член программного комитета от своего дома до дома своего товарища, округлённое вниз до целых.

## Пример

исходные данные	результат
3 10 10 20 20 10 20	13

## Исходный код:

```
n = int(input())
s = 0
x, y = [], []
for i in range(n):
    x1, y1 = map(int, input().split())
    x.append(x1)
    y.append(y1)
x.sort()
y.sort()

s = sum([(x[i+1]-x[i] + y[i+1]-y[i]) * (i+1) * (n-i-1) for i in range(n-1)])
print( (s * 2) // ((n-1)*n))
```

Заметим, что ходить в гости можно только по дорогам. Это сильно упрощает задачу, ведь тогда доходить до одной точки можно обходя другие. А координаты  $x$  и  $y$  никак не связаны между собой. Реализуется это так: читаем все  $x$  и  $y$ , добавляем в массивы и сортируем. Теперь остаётся пробежаться по ним в цикле, считая разницу между следующими  $x$  и  $y$  и перемножая на  $(i+1)*(n-(i+1))$ , так как именно столько раз пройдет конкретный путь. Теперь домножаем это на 2 и делим на количество путей – получаем среднюю длину пути.

Язык программирования: python

## Результат

ID	Дата	Автор	Задача	Язык	Результат проверки	№ теста	Время работы	Выделено памяти
<a href="#">10219287</a>	18:25:25 27 мар 2023	<a href="#">Maxim</a>	<a href="#">1726. Кто ходит в гости...</a>	PyPy 3.8 x64	Accepted		0.859	11 936 КБ

# №1521 «Военные учения 2»

Ограничение времени: 1.0 секунды

Ограничение памяти: 64 МБ

## Вступление

В ходе недавних военных учений (более подробно эта история рассказана в задаче [«Военные учения»](#)) министр обороны Советской Федерации товарищ Иванов имел возможность лично убедиться в блестящей боевой готовности солдат вверенной ему Советской Армии. Но одна вещь всё же продолжала беспокоить выдающегося военачальника. Прославленный генерал понимал, что была продемонстрирована лишь физическая подготовка солдат. Теперь настало время организовать очередные учения и проверить интеллектуальные способности личного состава.

Генерал Шульман, вновь назначенный ответственным за проведение учений, пожертвовал все выделенные деньги бедным и с чистой совестью лёг спать. Во сне генералу явился учебник по тактике и изложил схему, руководствуясь которой можно провести учения совершенно бесплатно.

## Задача

В соответствии с этой схемой учения делятся на  $N$  раундов, в течение которых  $N$  солдат, последовательно пронумерованных от 1 до  $N$ , маршируют друг за другом по кругу, т.е. первый следует за вторым, второй за третьим, ...,  $(N-1)$ -й за  $N$ -м, а  $N$ -й за первым. В каждом раунде очередной солдат выбывает из круга и идёт чистить унитазы, а оставшиеся продолжают маршировать. В очередном раунде выбывает солдат, марширующий на  $K$  позиций впереди выбывшего на предыдущем раунде. В первом раунде выбывает солдат с номером  $K$ .

Разумеется, г-н Шульман не питал никаких надежд на то, что солдаты в состоянии сами определить очерёдность выбывания из круга. «Эти неучи даже траву не могут ровно покрасить», – фыркнул он и отправился за помощью к прапорщику Шкурко.

## Исходные данные

Единственная строка содержит целые числа  $N$  ( $1 \leq N \leq 100000$ ) и  $K$  ( $1 \leq K \leq N$ ).

## Результат

Вывести через пробел номера солдат в порядке их выбывания из круга.

## Пример

исходные данные	результат
5 3	3 1 5 2 4

Исходный код:

```
def remove(v, l, r, pos):
    while l != r:
        tree[v][0] -= 1
        if pos <= tree[v * 2][0]:
            r = (l + r) // 2
            v = v * 2
        else:
            pos -= tree[v * 2][0]
            l = (l + r) // 2 + 1
            v = v * 2 + 1
    tree[v][0] -= 1
    return tree[v][1]

def build(v, l, r):
    if l == r:
        tree[v] = [1, l + 1]
    else:
        mid = (l + r) // 2
        build(v * 2, l, mid)
        build(v * 2 + 1, mid + 1, r)
        tree[v][0] = tree[v * 2][0] + tree[v * 2 + 1][0]
```

```

n, k = map(int, input().split())
tree = [[0] * 2 for i in range(4 * n)]
build(1, 0, n - 1)
current = k
for i in range(n):
    print(remove(1, 1, n, current), end=" ")
    current = (current - 1 + k) % (n - 1 - i)
    if current == 0: current = n - 1 - i

```

Для решения использовалась структура данных дерево отрезков. Для начала идёт его построение по информации о количестве солдат. Далее мы последовательно удаляем из него k солдат, в дереве оно ищется по алгоритму Binary Search Tree и удаляется. Тем временем переменная current меняет свое значение, идя «по кругу».

Язык программирования: python

Результат:

ID	Дата	Автор	Задача	Язык	Результат проверки	№ теста	Время работы	Выделено памяти
<a href="#">10219502</a>	20:52:51 27 мар 2023	<a href="#">Maxim</a>	<a href="#">1521. Военные учения 2</a>	PyPy 3.8 x64	Accepted		0.359	11 024 КБ

## Вывод

В результате проделанной работы были успешно решены 3 задачи, используя язык программирования python. Работа программ выдает правильный ответ, а также удовлетворяет условиям задачи (проходит по времени и памяти).