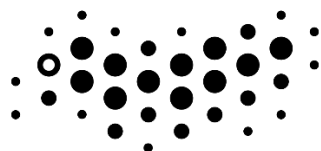


Санкт-Петербургский национальный  
исследовательский университет ИТМО

Факультет систем управления и робототехники



**ITMO UNIVERSITY**

# Алгоритмы и структуры данных. Отчет №2.

Выполнил студент гр. R32362

Лаптев Максим Сергеевич

Санкт-Петербург 2023

## **Содержание**

1. Цель
2. Задача №1155 – Дуоны
3. Задача №1207 – Медиана на плоскости
4. Задача №1322 – Шпион
5. Вывод

## **Цель**

Решить данные задачи, написав код, работа которого удовлетворяет условиям

# №1155 «Дуоны»

Ограничение времени: 0.5 секунды

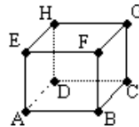
Ограничение памяти: 64 МБ

*Архангел по науке докладывает:*

*— Господи, эти физики там, внизу, — они открыли ещё одну элементарную частицу!*

*— Хорошо, добавим параметр в Общее Уравнение Вселенной.*

С развитием техники физики находят всё новые и новые элементарные частицы, с непонятными и даже загадочными свойствами. Многие слышали про мюоны, глюоны, странные кварки и прочую нечисть. Недавно были обнаружены элементарные частицы дуоны. Эти частицы названы так потому, что учёным удаётся создавать или аннигилировать их только парами. Кстати, от дуонов одни неприятности, поэтому от них стараются избавляться до начала экспериментов. Помогите физикам избавиться от дуонов в их установке.



Экспериментальная установка состоит из восьми камер, которые расположены в вершинах куба. Камеры промаркированы латинскими буквами A, B, C, ..., H. Технически возможно создать, или наоборот, аннигилировать, два дуона, находящихся в смежных камерах. Вам нужно автоматизировать процесс удаления дуонов из установки.

## Исходные данные

В единственной строке даны восемь целых чисел в пределах от 0 до 100, описывающих количество дуонов в камерах установки (сначала в камере A, потом в B, и т.д.).

## Результат

Выведите последовательность действий для удаления всех дуонов или слово «IMPOSSIBLE», если это невозможно. Каждое действие должно быть описано в отдельной строке, в следующем формате: маркер первой камеры, маркер второй (смежной с первой), далее плюс либо минус (создать или аннигилировать пару дуонов). Количество действий в последовательности не должно превосходить 1000.

## Примеры

исходные данные	результат
1 0 1 0 3 1 0 0	EF - EA - AD + AE - DC -
0 1 0 1 2 3 2 2	IMPOSSIBLE

## Исходный код:

```
cube = [int(j) for j in input().split()]
neighbours = {
    0: [1, 3, 4],
    1: [0, 2, 5],
    2: [1, 3, 6],
    3: [0, 2, 7],
    4: [0, 5, 7],
    5: [1, 4, 6],
    6: [2, 5, 7],
    7: [3, 4, 6]
}
```

```

letters = 'ABCDEFGH'
to_be_printed = []

# Удаление смежных
for i in range(8):
    if cube[i] > 0:
        for j in range(3):
            neighbour = neighbours[i][j]
            minimum = min(cube[neighbour], cube[i])
            if minimum:
                cube[i] -= minimum
                cube[neighbour] -= minimum
                for _ in range(minimum): to_be_printed.append(letters[i] +
letters[neighbour] + '-')

# Удаление по главной диагонали
diagonal = {0: 6, 1: 7, 2: 4, 3: 5, 4: 2, 5: 3, 6: 0, 7: 1}
for i in range(8):
    minimum = min(cube[diagonal[i]], cube[i])
    cube[i] -= minimum
    cube[diagonal[i]] -= minimum
    first_neighbour = neighbours[i][0]
    for el in neighbours[first_neighbour]:
        if el != i:
            second_neighbour = el
            break
    for j in range(minimum):
        to_be_printed.append(letters[first_neighbour] +
letters[second_neighbour] + '+')
        to_be_printed.append(letters[i] + letters[first_neighbour] + '-')
        to_be_printed.append(letters[diagonal[i]] + letters[second_neighbour]
+ '-')

if sum(cube) != 0:
    print("IMPOSSIBLE")
else:
    for el in to_be_printed:
        print(el)

```

Первым делом происходит удаление смежных дуонов, далее по главной диагонали путём добавления в любые смежные между ними вершины дуонов. Наконец остаются лишь по диагоналям, их удалить нельзя. Поэтому если сумма всего куба не равна нулю, то ответ – impossible.

Язык программирования: python

## Результат

ID	Дата	Автор	Задача	Язык	Результат проверки	№ теста	Время работы	Выделено памяти
<a href="#">10172057</a>	00:17:10 17 фев 2023	<a href="#">Maxim</a>	<a href="#">1155. Дуоны</a>	Python 3.8 x64	Accepted		0.078	420 КБ

## №1207 «Медиана на плоскости»

Ограничение времени: 0.5 секунды

Ограничение памяти: 64 МБ

На плоскости находятся  $N$  точек ( $N$  чётно). Никакие три точки не лежат на одной прямой. Ваша задача — выбрать две точки так, что прямая линия, проходящая через них, делит множество точек на две части одинакового размера.

### Исходные данные

Первая строка содержит целое число  $N$  ( $4 \leq N \leq 10\,000$ ). Каждая из следующих  $N$  строк содержит пары целых чисел  $x_i, y_i$  ( $-10^6 \leq x_i, y_i \leq 10^6$ ) — координаты  $i$ -й точки.

### Результат

Выведите номера выбранных точек.

### Пример

исходные данные	результат
4 0 0 1 0 0 1 1 1	1 4

### Исходный код:

```
from math import atan, pi

n = int(input())
s = []
x1, y1 = 9999999, 9999999
for i in range(n):
    x, y = map(int, input().split())
    if y < y1 or (y == y1 and x < x1):
        x1 = x
        y1 = y
        id = i
    s.append((x, y))
angles = []
for j, el in enumerate(s):
    if el[0] == x1 and el[1] == y1: continue
    elif el[0] == x1: angle = pi/2
    elif el[1] == y1: angle = 0
    else:
        angle = atan((el[1] - y1) / (el[0] - x1))
        if angle < 0:
            angle += 2 * pi
    angles.append((angle, j))
angles.sort()
print(id+1, angles[n//2-1][1] + 1)
```

Необходимо поделить множество точек на 2 подмножества одинакового размера. Для начала найдем самую крайнюю точку внизу слева, примем ее за точку отсчёта в полярной системе координат. Пройдемся по точкам еще раз и для

каждой посчитаем тангенс угла между рассматриваемой и началом координат. Теперь отсортируем по углу и возьмем середину – получим ответ.

Язык программирования: python

## Результат

ID	Дата	Автор	Задача	Язык	Результат проверки	№ теста	Время работы	Выделено памяти
<a href="#">10172133</a>	02:01:45 17 фев 2023	<a href="#">Maxim</a>	<a href="#">1207. Медиана на плоскости</a>	Python 3.8 x64	Accepted		0.125	2 956 КБ

# №1322 «Шпион»

Ограничение времени: 0.25 секунды

Ограничение памяти: 64 МБ

Спецслужбы обнаружили действующего иностранного агента. Шпиона то есть. Установили наблюдение и выяснили, что каждую неделю он через Интернет посылает кому-то странные нечитаемые тексты. Чтобы выяснить, к какой информации получил доступ шпион, требуется расшифровать информацию. Сотрудники спецслужб проникли в квартиру разведчика, изучили шифрующее устройство и выяснили принцип его работы.

На вход устройства подается строка текста  $S_1 = s_1s_2...s_N$ . Получив ее, устройство строит все циклические перестановки этой строки, то есть  $S_2 = s_2s_3...s_Ns_1$ , ...,  $S_N = s_Ns_1s_2...s_{N-1}$ . Затем множество строк  $S_1, S_2, ..., S_N$  сортируется лексикографически по возрастанию. И в этом порядке строчки выписываются в столбец, одна под другой. Получается таблица размером  $N \times N$ . В какой-то строке  $K$  этой таблицы находится исходное слово. Номер этой строки вместе с последним столбцом устройство и выдает на выход.

Например, если исходное слово  $S_1 = \text{abracadabra}$ , то таблица имеет такой вид:

1. aabracadabr =  $S_{11}$
2. abraabracad =  $S_8$
3. abracadabra =  $S_1$
4. acadabraabr =  $S_4$
5. adabraabrac =  $S_6$
6. braabracada =  $S_9$
7. bracadabraa =  $S_2$
8. cadabraabra =  $S_5$
9. dabraabraca =  $S_7$
10. raabracadab =  $S_{10}$
11. racadabraab =  $S_3$

И результатом работы устройства является число 3 и строка rdarcaaabb.

Это все, что известно про шифрующее устройство. А вот дешифрующего устройства не нашли. Но поскольку заведомо известно, что декодировать информацию можно (а иначе зачем же ее передавать?), Вам предложили помочь в борьбе с хищениями секретов и придумать алгоритм для дешифровки сообщений. А заодно и реализовать дешифратор.

## Исходные данные

В первой и второй строках находятся соответственно целое число и строка, возвращаемые шифратором. Длина строки и число не превосходят 100000. Строка содержит лишь следующие символы: a-z, A-Z, символ подчеркивания. Других символов в строке нет. Лексикографический порядок на множестве слов задается таким порядком символов:

## Исходные данные

В первой и второй строках находятся соответственно целое число и строка, возвращаемые шифратором. Длина строки и число не превосходят 100000. Строка содержит лишь следующие символы: a-z, A-Z, символ подчеркивания. Других символов в строке нет. Лексикографический порядок на множестве слов задается таким порядком символов:

ABCDEFGHIJKLMNOPQRSTUVWXYZ\_abcdefghijklmnopqrstuvwxyz

Символы здесь выписаны в порядке возрастания.

## Результат

Выведите декодированное сообщение в единственной строке.

## Пример

исходные данные	результат
3 rdarcaaabb	abracadabra

## Исходный код:

```
#include <algorithm>
#include <iostream>

using namespace std;

int main() {
    int n, len;
    string s;
    cin >> n >> s;
    len = s.size();
    pair<int, char> dict[len];

    for (int i = 0; i < len; i++){
        dict[i].first = i;
        dict[i].second = s[i];
    }
    sort(dict, dict + len, [](const pair<int, char>& x1, const pair<int, char>& x2) -> bool {
        return x1.second < x2.second;
    });

    n = n - 1;
    for (int i = 0; i < len; i++) {
        n = dict[n].first;
        cout << s[n];
    }

    return 0;
}
```

Сначала я хотел использовать «Burrows-Wheeler Data Compression Algorithm», однако он все никак не проходил по времени. Спустя какое-то время я догадался до более быстрого решения, но и тут на python он не проходил, и я пересел на C++.

Суть в том, что сначала мы заполняем словарь значениями (индекс, символ) и сортируем его. В результате получаем список из предыдущих значений последовательности. Теперь пройдясь по нему, начиная с заданного индекса, можно получить ответ.

Язык программирования: C++

## Результат:

ID	Дата	Автор	Задача	Язык	Результат проверки	№ теста	Время работы	Выделено памяти
<a href="#">10176221</a>	22:21:04 20 фев 2023	<a href="#">Maxim</a>	<a href="#">1322. Шпион</a>	G++ 9.2 x64	Accepted		0.015	2 184 КБ



## **Вывод**

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