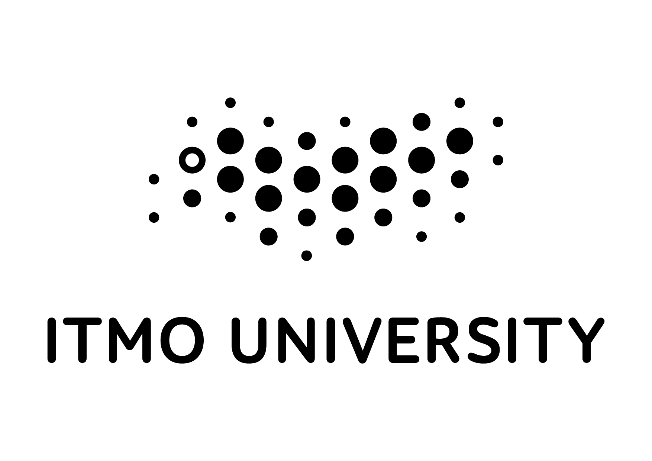
Санкт-Петербургский национальный исследовательский университет ИТМО

Факультет систем управления и робототехники



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»**

Изображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как стол

Автоматически созданное описание

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



**№1207** **«Медиана на плоскости»**

Изображение выглядит как текст

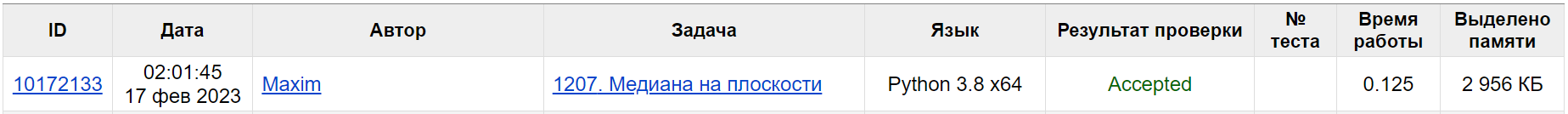
Автоматически созданное описание

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

**№1322 «Шпион»**

Изображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как текст

Автоматически созданное описание

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: С++

Результат:



**Вывод**

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