Санкт-Петербургский национальный исследовательский университет ИТМО

**Отчёт по лабораторной работе №1155**

**по дисциплине «Алгоритмы и структуры данных»**

Выполнил:

студент гр. R3242

Ворков Никита

Преподаватель:

Тропченко А. А.

Санкт-Петербург

2021

**Цель работы:**

Приблизиться к зачету по алгосам

**Текст задачи:**

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.

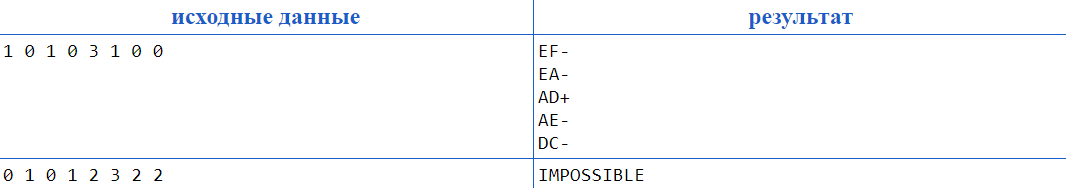
**Код задачи:**

#include <iostream>  
#include <vector>  
  
*//удаление максимального кол-во дуонов в соседних вершинах*void del(std::vector<int> &nums, int i1, int i2)  
{  
 while (std::min(nums[i1], nums[i2])) {  
 std::cout << char(int('A') + i1) << char(int('A') + i2) << '-' << std::endl;  
 --nums[i1];  
 --nums[i2];  
 }  
}  
  
*//добавление dif дуонов в соседние вершины*void add(std::vector<int> &nums, int i1, int i2, int dif)  
{  
 for (int i = 0; i < dif; ++i) {  
 std::cout << char(int('A') + i1) << char(int('A') + i2) << '+' << std::endl;  
 ++nums[i1];  
 ++nums[i2];  
 }  
}  
  
int main()  
{  
  
 std::vector<int> nums(8);  
 std::vector<int> kostyl\_vector = {1, 2, 3, 7, 4, 5, 1};  
  
 for (int i = 0; i < 8; ++i)  
 std::cin >> nums[i];  
  
 if (nums[0] + nums[2] + nums[5] + nums[7] != nums[1] + nums[3] + nums[4] + nums[6]) {  
 std::cout << "IMPOSSIBLE";  
 return 0;  
 }  
  
 *//зануление вершины A* del(nums, 0, 4);  
 del(nums, 0, 3);  
 del(nums, 0, 1);  
  
 if (nums[0]) {  
 add(nums, 4, 5, nums[0]);  
 del(nums, 0, 4);  
 }  
  
 *//зануление вершины G* del(nums, 6, 5);  
 del(nums, 6, 7);  
 del(nums, 6, 2);  
  
 if (nums[6]) {  
 add(nums, 5, 4, nums[6]);  
 del(nums, 6, 5);  
 }  
  
 *//гоним дуоны по кругу B C D H E F* for (int i = 0; i < 5; ++i) {  
 del(nums, kostyl\_vector[i], kostyl\_vector[i + 1]);  
 if (nums[kostyl\_vector[i]]) {  
 add(nums, kostyl\_vector[i + 1], kostyl\_vector[i + 2], nums[kostyl\_vector[i]]);  
 del(nums, kostyl\_vector[i], kostyl\_vector[i + 1]);  
 }  
 }  
  
 return 0;  
}

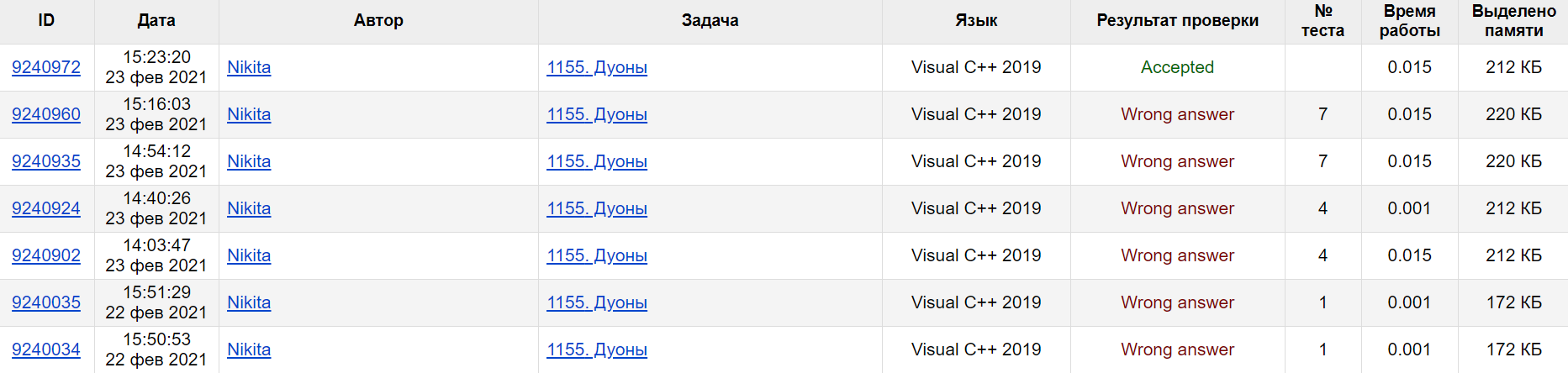
**Используемые алгоритмы и структуры данных:**

Суть алгоритма заключается в том, чтобы сначала занулить вершины A и G, а остальные вершины представить в виде многоугольника B, C, D, H, E, F. В данном многоугольнике последовательно рассматриваются соседние вершины. Если вершина i+1 больше вершины i, то уменьшаем обе вершины. Если вершина i+1 меньше вершины i, то увеличиваем вершины i+1 и i+2, а потом уменьшаем вершины i и i+1. В кубе все вершины можно разделить на 2 группы несмежных между собой. Если суммы вершин в таких группах не совпадают, то решения нет и наоборот.

**Входные и выходные данные:**



**Скрин из тимуса:**

****

**Вывод**

Потратив 6 часов на реализацию сложного алгоритма, я сделал вывод, что стоит сначала попробовать более простые алгоритмы. Это сохранит и нервы и психику.