

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»

Отчёт по лабораторной работе №6  
«Динамическое программирование. Задание 514. Freedom Trail»

Выполнила работу:  
Абаянцева Евгения  
Академическая группа №J3110  
Принято:  
...

Санкт-Петербург  
2024 г.

# Введение

Цель – написать код с использованием динамического программирования по задаче с сайта LeetCode.

Задача: в видеоигре Fallout 4 в задании «Дорога к свободе» игрокам необходимо добраться до металлического диска, называемого «Кольцо пути свободы», и использовать диск, чтобы произнести определенное ключевое слово, чтобы открыть дверь.

Если задана строка `ring`, представляющая код, выгравированный на внешнем кольце, и другая строка `key`, представляющая ключевое слово, которое необходимо написать, верните минимальное количество шагов для написания всех символов в ключевом слове .

Первоначально первый символ кольца выровнен по направлению "12:00". Вы должны написать все символы по `key` одному, вращая по `ring` часовой стрелке или против часовой стрелки, чтобы выровнять каждый символ клавиши строки по "12:00" направлению, а затем нажать центральную кнопку.

На этапе вращения кольца для произнесения ключевого символа `key[i]`:

1. Вы можете вращать кольцо по часовой стрелке или против часовой стрелки на одно место, что считается одним шагом . Конечная цель вращения – выровнять один из `ring` символов в "12:00" направлении, где этот символ должен быть равен `key[i]`.

2. Если символ `key[i]` был выровнен по "12:00" направлению, нажмите центральную кнопку для написания, что также считается одним шагом. После нажатия вы можете начать писать следующий символ в ключе (следующий этап). В противном случае вы закончили все написание.

## Теоретическая подготовка

Данная задача предполагает решение через динамическое программирование. Задача разбивается на повторяющиеся подзадачи (в случае задачи 514 нахождение минимальных путей от одной буквы в кольце до другой, а затем снова повторяется). Используемые типы данных для этой задачи – целые числа (тип `int`), строки (тип `string`), кортеж (тип `vector`), словарь (`unordered_map`).

## Реализация

Для задачи 514 требуются следующие библиотеки:

1. Библиотека `iostream` – эта основная библиотека C++ для ввода и вывода текста с консоли;
2. Библиотека `vector` – эта библиотека определяет вектор шаблона класса контейнера и несколько вспомогательных шаблонов;
3. Библиотека `unordered_map` – эта библиотека определяет шаблон класса `unordered_map`;
4. Библиотека `algorithm` – эта библиотека определяет функции шаблона контейнера стандартной библиотеки C++, которые выполняют алгоритмы;
5. Библиотека `climits` – такая библиотека позволяет автоматически указывать максимальные и минимальные значения типов данных.

Реализация была сделана путём оптимизации рекурсии: вместо того, чтобы каждый раз пробегать по одной и той же функции, было выбрано динамическое программирование, которое на каждом шагу цикла позиции элемента ставило минимальное количество вращений (шагов) до следующего элемента. В конце одного прохода считается минимальный шаг по всем таким позициям, после чего делается ещё один шаг, то есть переход к следующей букве.

Особенность реализации: повторяющаяся подзадача в виде проверки минимального элемента на каждом шагу алгоритма.

## Экспериментальная часть

1. Подсчёт памяти: 1 словарь (40 байт), 1 кортеж (16 байт), 1 `char` (1 байт), 1 `int` (4 байт) – в сумме 65 байт. Два `int` (8 байт). Всего в сумме – 73 байта.

2. Асимптотика:  $O(n \cdot m)$ .

## Заключение

В ходе выполнения этой задачи было реализовано динамическое программирование. Цель работы была достигнута путём проверки кода на сайте LeetCode.

В качестве дальнейшего исследования можно предложить оптимизацию алгоритма с точки зрения уменьшения требуемой памяти и времени на исполнение алгоритма.

# Приложения

```
1 #include <iostream>
2 #include <vector>
3 #include <unordered_map>
4 #include <algorithm>
5 #include <climits>
6
7 class Solution {
8 public:
9     int findRotateSteps(std::string ring, std::string key) {
10         std::unordered_map<char, std::vector<int>> ltr_indexes;
11         // std::cout << sizeof(ltr_indexes) << std::endl;
12         ltr_indexes['#'].push_back(0);
13         int ln = ring.size();
14         // std::cout << sizeof(ln) << std::endl;
15
16         for (int idx = 0; idx < ln; idx++) {
17             ltr_indexes[ring[idx]].push_back(idx);
18         } // f(n)
19
20         std::vector<int> l_lens(ln, 0);
21         // std::cout << sizeof(l_lens) << std::endl;
22
23         char prev_ltr = '#';
24         for (char ltr : key) {
25             for (int pos : ltr_indexes[ltr]) {
26                 std::vector<int> all_variants;
27                 // std::cout << sizeof(all_variants) << std::endl;
28
29                 for (int prev_pos : ltr_indexes[prev_ltr]) {
30                     int clk_w = std::abs(prev_pos - pos);
31                     // std::cout << sizeof(clk_w) << std::endl;
32                     int a_clk = ln - clk_w;
33                     // std::cout << sizeof(a_clk) << std::endl;
34                     all_variants.push_back(std::min(clk_w, a_clk) + l_lens[prev_pos]);
35                 } // f(n^3)
36
37                 l_lens[pos] = *std::min_element(all_variants.begin(), all_variants.end()
38 );
39             }
40             prev_ltr = ltr;
41         }
42
43         int min_len = 9 * 10000;
44         // std::cout << sizeof(min_len) << std::endl;
45
46         for (int pos : ltr_indexes[prev_ltr]) {
47             std::vector<int> numbers{ min_len, l_lens[pos] };
48             min_len = *std::min_element(numbers.begin(), numbers.end());
49         } // f(n)
50
51         std::cout << sizeof(min_len) + sizeof(l_lens) + sizeof(ltr_indexes) + sizeof
52 (ln) + 1 << std::endl;
53
54         return min_len + key.size();
55     }
56 };
57
58 int main() {
```

```

57 Solution x;
58 std::cout << x.findRotateSteps("aebcdgb", "abc") << std::endl;
59 return 0;
60 }

```

Листинг 1 – код файла lab-6.cpp

Прохождение всех тестов на сайте LeetCode:

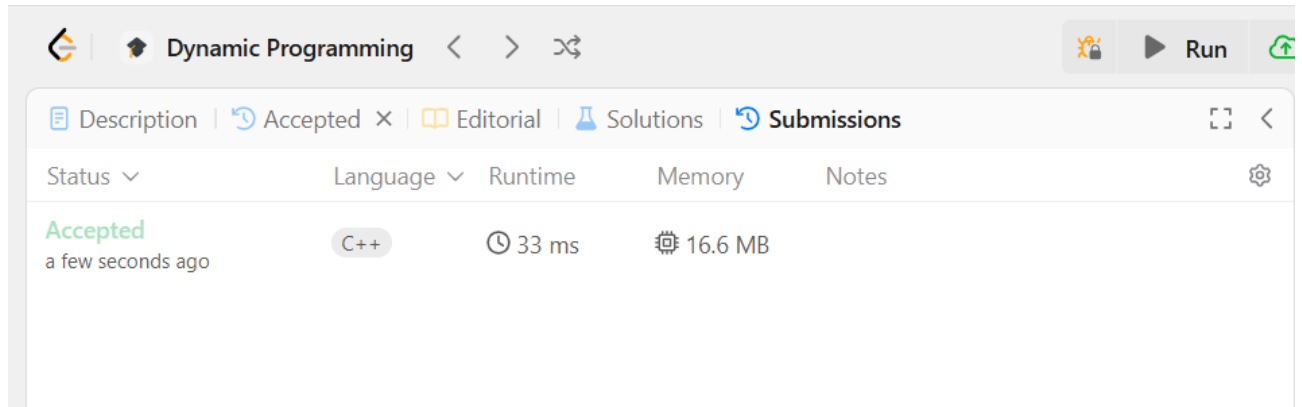


Рис. 1 – основные тесты