

**Министерство науки и высшего образования Российской
Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

Факультет цифровых трансформаций

Дисциплина:
«Алгоритмы и структуры данных»

Практическая работа №6
«Динамическое программирование»

Выполнила:
Абаянцева Е. Ю., студент группы J3110

Санкт-Петербург
2024 г.

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. Подсчёт асимптотики – сложность $O(n \cdot m)$.
3. Код:

```

lab-6.cpp Solution findRotateSteps(std::string ring, std::string key)
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         ltr_indexes['#'].push_back(0);
12         int ln = ring.size();
13
14         for (int idx = 0; idx < ln; idx++) {
15             ltr_indexes[ring[idx]].push_back(idx);
16         }
17
18         std::vector<int> l_lens(ln, 0);
19
20         char prev_ltr = '#';
21         for (char ltr : key) {
22             for (int pos : ltr_indexes[ltr]) {
23                 std::vector<int> all_variants;
24
25                 for (int prev_pos : ltr_indexes[prev_ltr]) {
26                     int clk_w = std::abs(prev_pos - pos);
27                     int a_clk = ln - clk_w;
28                     all_variants.push_back(std::min(clk_w, a_clk) + l_lens[prev_pos]);
29                 }
30
31                 l_lens[pos] = *std::min_element(all_variants.begin(), all_variants.end());
32             }
33             prev_ltr = ltr;
34         }
35
36         int min_len = 9 * 10000;
37
38         for (int pos : ltr_indexes[prev_ltr]) {
39             std::vector<int> numbers{ min_len, l_lens[pos] };
40             min_len = *std::min_element(numbers.begin(), numbers.end());
41         }
42
43         return min_len + key.size();
44     }

```

Рис. 1: КОД

4. Прохождение всех локальных тестов:

(а) Тест 1:

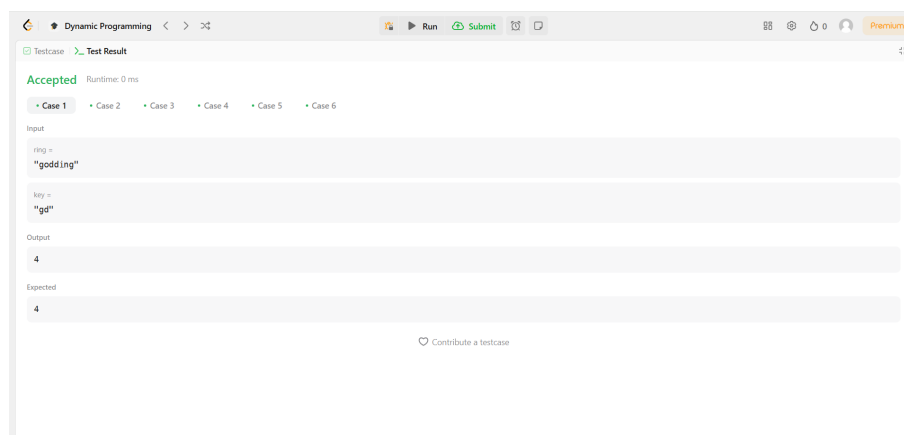


Рис. 2: ТЕСТ 1

(b) Тест 2:

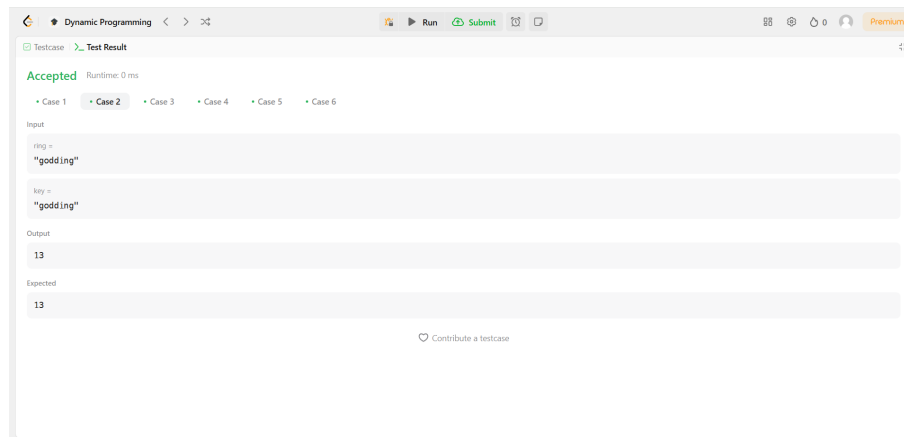


Рис. 3: ТЕСТ 2

(c) Тест 3:

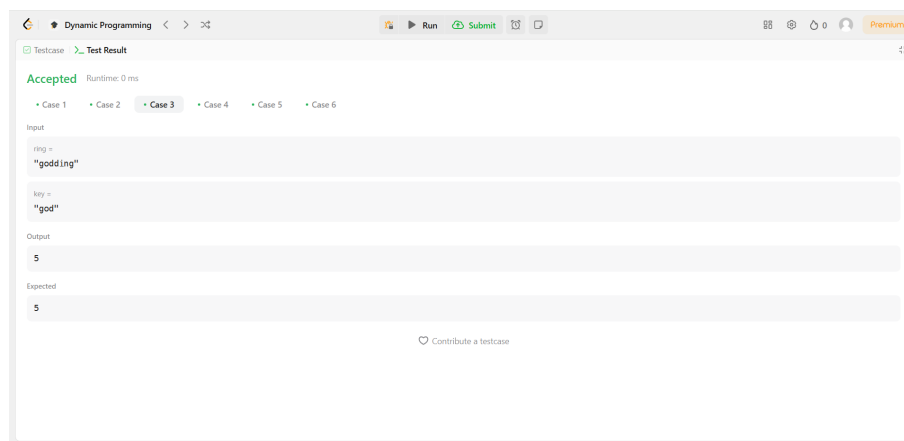


Рис. 4: ТЕСТ 3

(d) Тест 4:

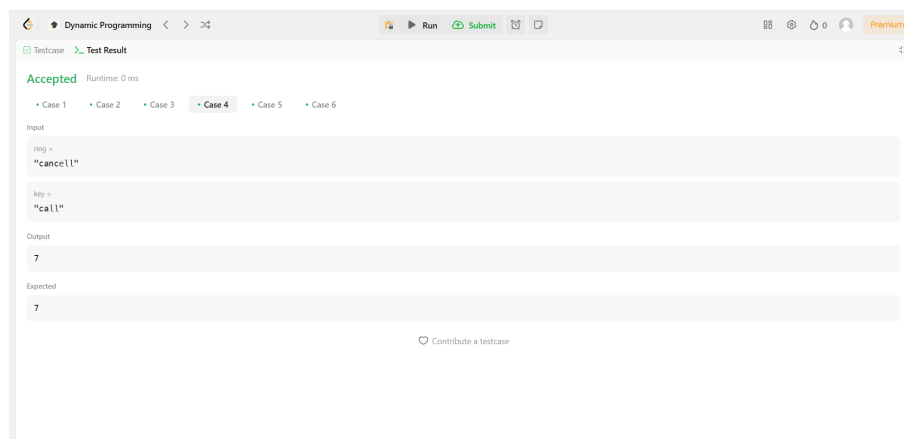


Рис. 5: ТЕСТ 4

(e) Тест 5:

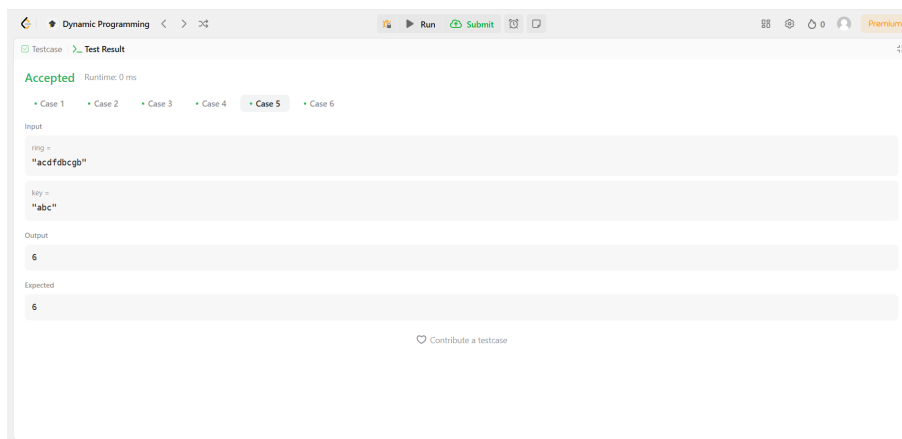


Рис. 6: ТЕСТ 5

(f) Тест 6:

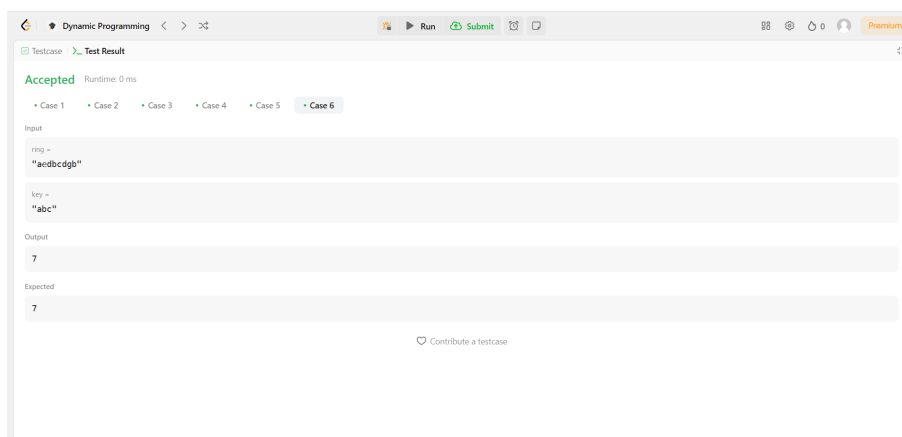


Рис. 7: ТЕСТ 6

5. Прохождение основных тестов:

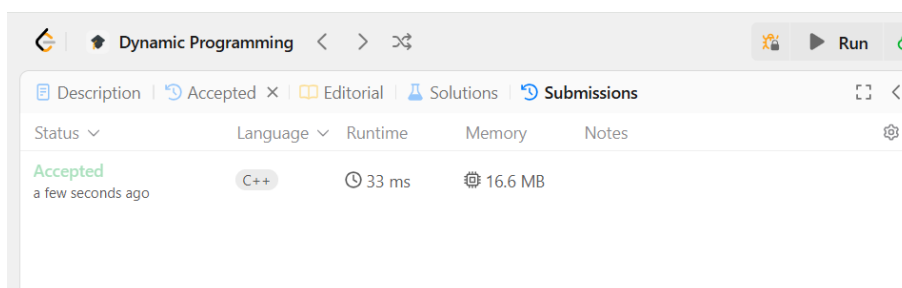


Рис. 8: ОСНОВНЫЕ ТЕСТЫ

6. В этой задаче требуется использовать динамическое программирование, так как перебор всех элементов в кольце (идя по порядку или поэлементно) сильно перегружает алгоритм и требует больше памяти, чем могло бы потребоваться.