

Задача F

1) Алгоритм

Для решения задачи мы проходим по ленте и проверяем, можно ли выделить слова из заданного текста поочередно. Была реализована функция `main_algo`, которая:

1. Проходит по строке `text` и ищет в ней последовательное вхождение каждого слова из заданного списка `words`.
2. Проверяет, что после каждого найденного слова между ним и следующим словом остается хотя бы одна закрашенная буква.
3. Проверяет, что все слова из списка присутствуют в строке в указанном порядке.

Функция `main_algo` принимает два аргумента:

- `words` — список слов, который должен быть на ленте.
- `text` — строка, представляющая ленту, на которой нужно найти эти слова в нужном порядке.

Алгоритм работает следующим образом:

- Инициализируются два указателя: `ptr_word` для текущего слова из списка `words` и `i` для текущей позиции в строке `text`.
- Цикл продолжается, пока `i` меньше длины строки `text` и пока `ptr_word` не достиг конца списка слов. Если текущее слово может быть найдено в строке начиная с текущей позиции `i`, то увеличивается указатель `i` на длину слова, а указатель `ptr_word` переходит к следующему слову. После нахождения каждого слова увеличивается указатель `i`, чтобы убедиться, что между словами есть хотя бы одна закрашенная буква.
- По завершению работы цикла проверяется, что все слова из списка были найдены.

2) Сложность решения

Алгоритм выполняет линейный проход по строке `text` и списку слов `words`. Таким образом, временная сложность решения составляет $O(N+M)$, где N — длина строки `text`, а M — суммарная длина всех слов в списке `words`.

Памятная сложность: используется дополнительная память для хранения списка слов и указателей, что требует $O(M)$ памяти.

Алгоритм гарантирует, что все слова из списка будут найдены в строке `text` в правильном порядке. Проверка наличия хотя бы одной закрашенной буквы между словами выполняется с помощью увеличения указателя `i` после нахождения каждого слова. Если указатель `ptr_word` достигает конца списка `words` к завершению работы цикла, это означает, что все слова были найдены в нужном порядке.

3) Распечатка кода

```
#include <iostream>
#include <string>
#include <vector>

using namespace std;

bool main_algo(const std::vector<std::string>& words, const
std::string& text) {
    size_t ptr_word = 0;
    size_t i = 0;
    while (i < text.size() && ptr_word < words.size()) {
        if (i + words[ptr_word].size() <= text.size() &&
            text.substr(i, words[ptr_word].size()) ==
words[ptr_word]) {
            i += words[ptr_word].size();
            ptr_word++;
            if (ptr_word < words.size()) {
                if (i >= text.size()) {
                    return false;
                }
                i++;
            }
        } else {
            i++;
        }
    }
    while (i < text.size()) {
        i++;
    }
    return ptr_word == words.size();
}

int main() {
    size_t n_digit, m_digit;
    std::cin >> n_digit >> m_digit;
    std::string s_str;
    std::cin >> s_str;
```

```
std::vector<std::string> words(m_digit);  
for (size_t i = 0; i < m_digit; ++i) {  
    std::cin >> words[i];  
}  
  
if (main_algo(words, s_str)) {  
    std::cout << "YES" << std::endl;  
} else {  
    std::cout << "NO" << std::endl;  
}  
return 0;  
}
```