ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»

Отчёт по лабораторной работе № 6

«Динамическое программирование»

Выполнил работу

Вагин Арсений

Академическая группа №J3112

Принято

Должность, звание Фамилия Имя преподавателя

Санкт-Петербург

2024

**1. Подсчет по памяти**

Для оценки памяти, используемой решением, важно учитывать несколько компонентов:

1. Исходная строка s: Хранится как объект строки O(n), где n — длина строки.
2. Таблица мемоизации dp: Матрица размером n×n, где каждый элемент хранит int, что потребляет O(n^2) памяти.
3. Временные переменные: Несколько переменных в рекурсивной функции, включая указатели и временные значения. Память под стек рекурсии ограничена глубиной вызовов, которая максимум равна O(n).

Итог: Общая используемая память составляет O(n^2).

**2. Подсчет асимптотики**

1. Цикл в рекурсивной функции:
   * Основной цикл: for (int k = start + 1; k <= end; ++k)
   * Длина этого цикла пропорциональна O(n).
   * Вложенные вызовы рекурсии также проверяют подстроки, разделенные точкой k, что создает O(n^2) вызовов.
2. Общая сложность алгоритма:
   * Сложность полного решения: O(n^3), где n — длина строки.
   * O(n^2) диапазонов обрабатывается по O(n) операций для каждого диапазона.

**3. Добротный код**

**Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание**

1. **Скрины, что задача прошла все тесты**

****

**Изображение выглядит как текст, Шрифт, снимок экрана, Графика

Автоматически созданное описание**

**5. Пояснение, почему нужно ДП**

Почему динамическое программирование (ДП):

* Задача требует оптимального решения с минимальным количеством шагов. Это подразумевает пересчет множества перекрывающихся подзадач.
* Каждый диапазон (start, end) может пересекаться с другими диапазонами из-за возможности объединения символов.
* Хранение результатов для подзадач в таблице dp предотвращает избыточные вычисления, что значительно ускоряет алгоритм.

Преимущества ДП:

* Обеспечивает вычисление оптимального решения за O(n^3) вместо экспоненциальной сложности O(2^n) без мемоизации.
* Применим для задач, где решение можно разбить на независимые подзадачи (разбиение строки на диапазоны).