ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»

Отчёт по лабораторной работе № 7

«Жадные алгоритмы»

Выполнил работу

Бондаренко Анна

Академическая группа №J3112

Принято

Дунаев Максим

Санкт-Петербург

2024

**Структура отчёта:**

1. Введение

В данной задаче необходимо выбрать n кусочков пиццы из массива slices, состоящего из 3n элементов, чтобы их сумма была максимальной. При этом соседние кусочки нельзя выбирать, а первый и последний элементы массива считаются соседними (массив циклический).

1. Задачи

* Написать код на с++
* Подсчитать временную и пространственную сложность алгоритма
* Проанализировать, в чём заключается жадный подход

1. Теоретическая подготовка

**Динамическое программирование**:

Основная идея состоит в разбиении задачи на подзадачи и их решении с использованием уже полученных результатов.

Для подмассива мы определяем, какова максимальная сумма, если мы выбрали j элементов из первых i доступных.

Чтобы учесть цикличность массива (первый и последний элементы — соседи), мы разбиваем задачу на два случая:

* Рассматриваем массив без первого элемента.
* Рассматриваем массив без последнего элемента.
* Затем берём максимум из двух результатов.

**Жадный подход**:

Жадные алгоритмы предполагают выбор локально оптимального решения на каждом шаге, надеясь, что это приведёт к глобальному оптимуму.

В нашей задаче жадный шаг заключается в выборе между двумя случаями: исключение первого элемента или последнего. Это выбор, который кажется оптимальным локально.

**Исходный массив**slices:

Тип: vector<int>. Хранит размеры кусочков пиццы в виде целых чисел (для массива [8,9,8,6,1,1] каждый элемент slices[i] представляет размер i-го кусочка).

**Таблица динамического программирования**dp:

Тип: vector<vector<int>>.

Размер: m+1×n+1, где m — длина подмассива, n — количество выбираемых кусочков.

dp[i][j] хранит максимальную сумму, которую можно получить, выбрав j кусочков из первых i доступных.

**Переменные цикла и индексации**:

Тип: int. Используются для итерации по массивам и индексирования элементов.

**Промежуточные результаты**:

Тип: int. Хранят результаты вычислений, такие как максимальная сумма из двух случаев (case1 и case2).

1. Реализация

**Жадный шаг в решении:**

Главный жадный элемент состоит в выборе между двумя подмассивами:

* 1. Рассматриваем массив, исключив первый элемент.
  2. Рассматриваем массив, исключив последний элемент.

Этот выбор основан на максимизации результата (берём максимум из двух вариантов), что является примером жадного подхода. Однако, внутри каждого случая используется динамическое программирование, а не жадный выбор.

1. Экспериментальная часть

**1.Временная сложность**:

Основная часть вычислений выполняется в функции maxSum

В функции мы перебираем:

* + - m элементов подмассива (где m≈2nm≈2n).
    - n возможных выбираемых кусочков.

Заполнение таблицы dp занимает O(m×n).

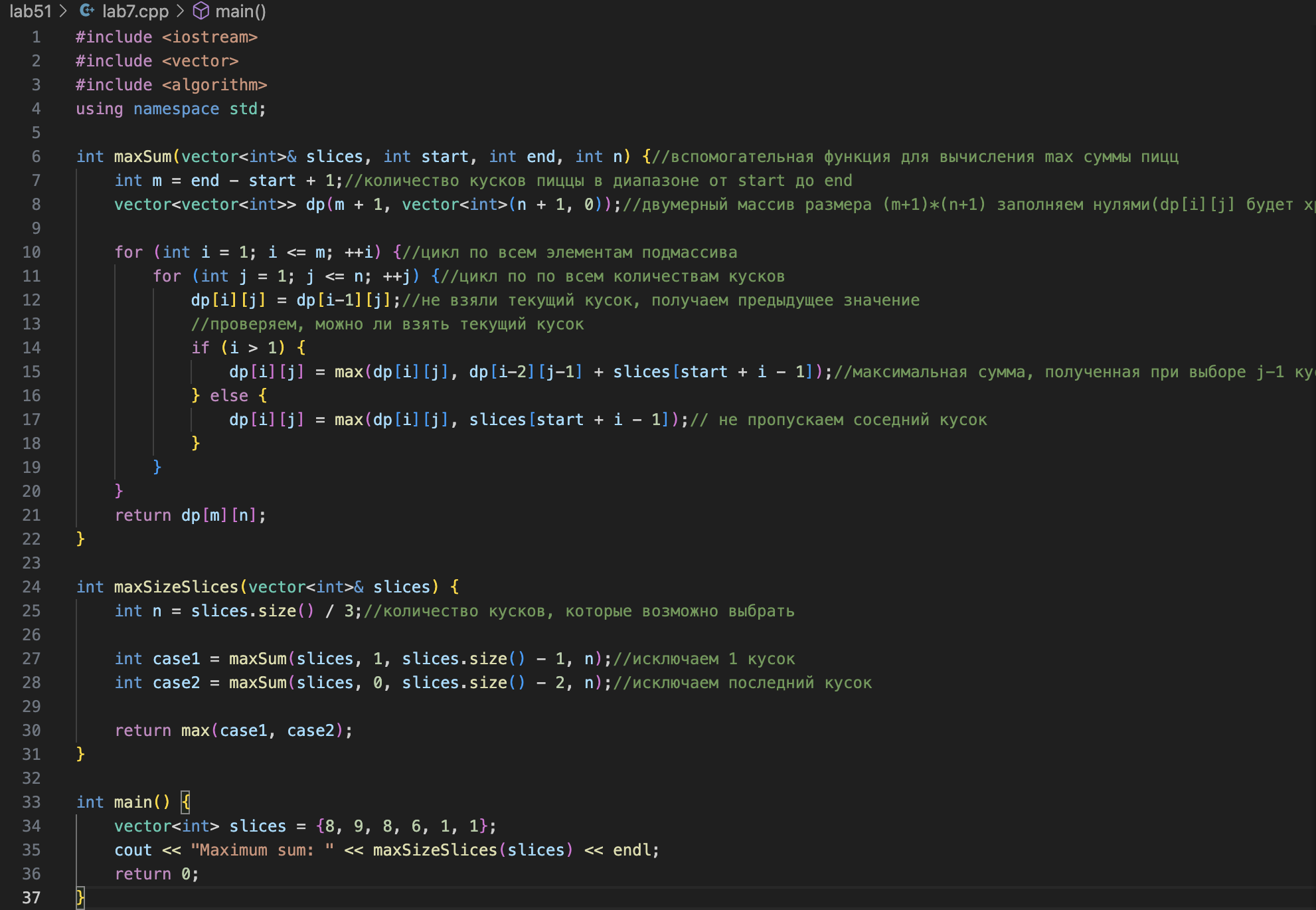
Так как maxSum вызывается дважды, общая временная сложность:O(2×m×n)=O(n^2)

1. **Пространственная сложность**:
   * Таблица dp имеет размер (m+1)×(n+1), где m≈2nm≈2n.
   * Пространственная сложность:O(m×n)=O(n^2)
   * Дополнительно используется незначительное количество памяти для переменных, поэтому итоговая пространственная сложность — O(n^2).
2. Заключение

В решении задачи использован в основном метод динамического программирования. Однако присутствует элемент жадного подхода при выборе между двумя основными случаями (исключение первого или последнего элемента массива). Динамическое программирование гарантирует правильный учёт всех возможных вариантов выбора, что невозможно при чисто жадном решении.

1. Приложения

ПРИЛОЖЕНИЕ А

Листинг кода