****

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

**«МИРЭА – Российский технологический университет»**

# РТУ МИРЭА

Отчет по выполнению практического задания №8.2

**Тема: «ОТДЕЛЬНЫЕ ВОПРОСЫ АЛГОРИТМИЗАЦИИ»**

Дисциплина: Структуры и алгоритмы обработки данных

Выполнил студент Кузнецов Л. А.

группа ИКБО-20-23

**Москва 2024СОДЕРЖАНИЕ**

[ЧАСТЬ 8.2………………………………………………………………………...3](#часть)

[Задание……………………………………………………………………………3](#задание)

[Вариант…………………………………………………………………………...3](#вариант)

[Решение задачи…………………………………………………………………..3](#решение)

[Тесты……………………………………………………………………………...8](#тесты)

[ВЫВОД………………………………………………………………………….10](#вывод)

[СПИСОК ИНФОРМАЦИОННЫХ ИСТОЧНИКОВ………………………....10](#список)

**ЧАСТЬ 8.2**

**Задание**

1. Разработать алгоритм решения задачи с применением метода, указанного в варианте и реализовать программу.

2. Оценить количество переборов при решении задачи стратегией «в лоб» - грубой силы. Сравнить с числом переборов при применении метода.

3. Оформить отчет в соответствии с требованиями документирования разработки ПО: Постановка задачи, Описание алгоритмов и подхода к решению, Код, результаты тестирования, Вывод.

**Вариант**

18 вариант - разработать процедуру оптимального способа расстановки скобок в произведении последовательности матриц, размеры которых равны (5,10,3,12,5,50,6), чтобы количество скалярных умножений стало минимальным (максимальным) (“Жадный алгоритм”).

**Решение задачи**

Для начала стоит ознакомиться с самим жадным алгоритмом: он представляет из себя выбор наилучшего исхода на данный момент без анализа последствий сделанного выбора.

Так как мы работаем с матрицами и их скалярным умножением, то итоговый результат кол-ва умножений можно получить перемножив ряды первой матрицы на столбцы второго, что и будет использоваться в дальнейшей работе как основной метод оценки кол-ва перемножений.

Математическая модель: проходимся необходимое кол-во раз по всем имеющимся матрицам и определяем выше описанным методом кол-во перемножений. После этого проставляем скобки там, где нашли наименьшее/наибольшее кол-во перемножений. Повторяем столько раз, сколько матриц используется в самой задаче.

У моего метода решения задачи фиксированное значение проходов, а именно (формула 1)

(1)

Где *n* – кол-во матриц.

Стоит отметить, что *S* – кол-во проходов по всем элементам, а не кол-во повторений цикла, ведь кол-во повторений цикла тоже фиксированное, а именно *n* раз повторяется цикл.

Также можно аналитически определить кол-во повторений проверки всех возможных исходов «в лоб». В случае такого перебора нам придётся пройти приблизительно , что больше полученных значений перебора, используя “жадный алгоритм”.

Теперь же перейдём к описанию кода программы (рис. 1-4).

Начнём, пожалуй, с описания всех методов, необходимых для реализации алгоритма (рис. 1).

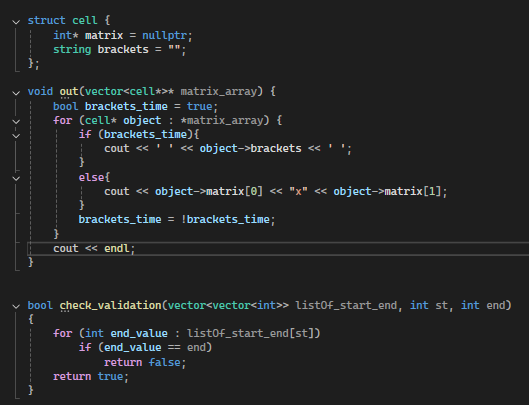


Рисунок 1 – Все необходимые для реализации алгоритма методы

На рис.1 представлена структура cell, хранящая в себе все данные о наших взаимодействиях с матрицами (саму матрицу и наличие скобок). Ещё есть метод out, который реализует вывод на экран всех элементов нашего уравнения. А check\_validation служит способом избежания повторной простановки скобок в одном и том же месте.

На рис.2-4 представлен сам код «Жадного алгоритма».

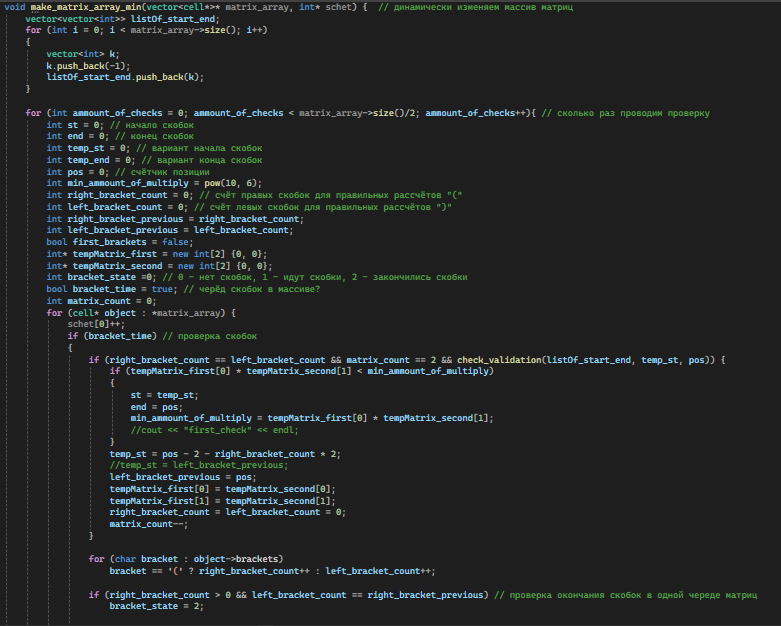


Рисунок 2 – 1-ая часть реализации «Жадного алгоритма»

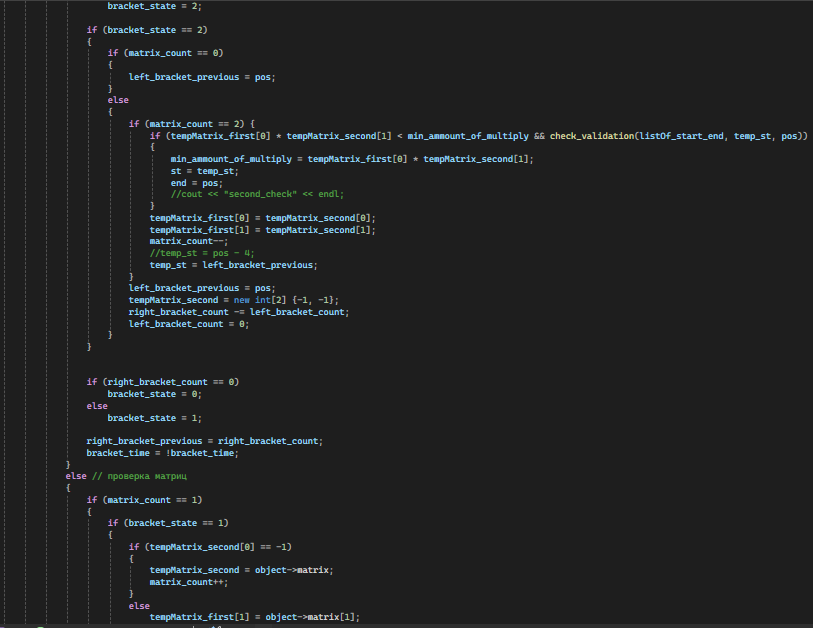


Рисунок 3 – 2-ая часть реализации «Жадного алгоритма»

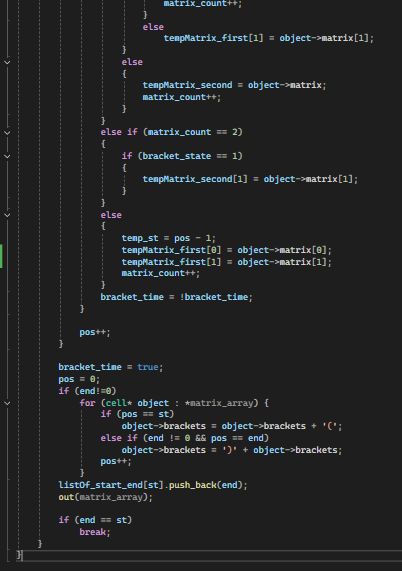


Рисунок 4 – 3-я часть реализации «Жадного алгоритма»

Алгоритм, представленный на рис.2-4 был описан ранее, однако это не отменяет того факта, что, несмотря на всю свою простоту трактовки, реализации представила из себя довольно трудоёмкий процесс, так пришлось обрабатывать множество ситуаций и дать программе такие указания, чтобы она была способна определить необходимый результат, опираясь исключительно на текущие значения. Это была самая большая и трудная часть работы.

Далее проиллюстрируем на рис.5 метод создания списка матриц с заданными ограничениями.

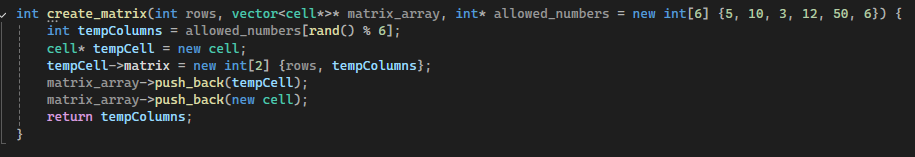


Рисунок 5 – Метод создания списка матриц с заданными ограничениями

В данном метода мы банально создаём случайным образом следующую матрицу в зависимости от кол-ва столбцов предыдущей (переменная названа rows из-за того, что в текущей матрице они примет смысл как раз-таки рядов, а не столбцов).

И последняя иллюстрация в текущем разделе – рис.6 реализация алгоритма работы программы.



Рисунок 6 – Реализация алгоритма работы программы

Алгоритм на рис.6 реализует простейший CLI и направляет пользователя на создание уравнения в соответствии с задачей.

**Тесты**

Теперь же приступим к тестам нашей программы (рис.7-10).

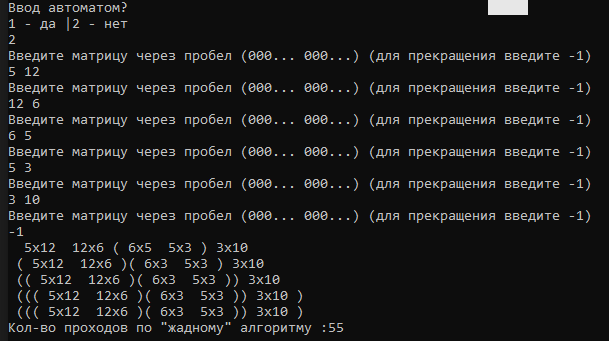


Рисунок 7 – 1-ый тест работы программы

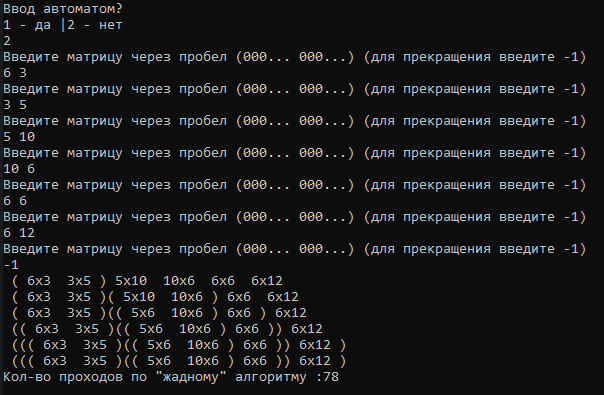


Рисунок 8 – 2-ый тест работы программы

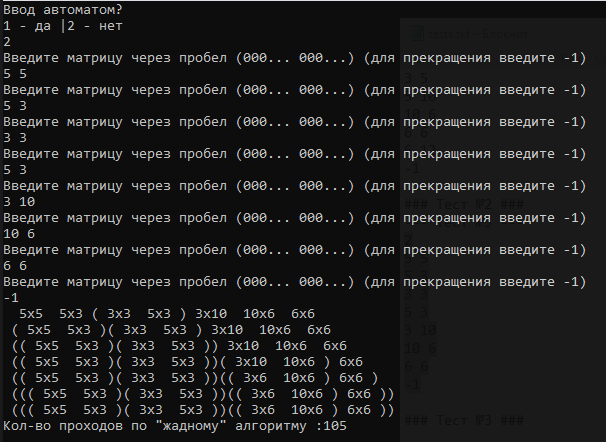


Рисунок 9 – 3-ый тест работы программы

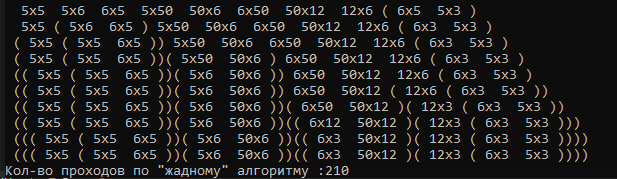


Рисунок 10 – 4-ый тест работы программы

Данные тесты были выбраны именно из-за того, что каждый из них образует неповторимую ситуацию, проверяющую на точность и стабильность нашу программу, ведь каждый раз, когда алгоритм работал во всех случаях, находился один, который выставлял условия, не рассматриваемые плохо реализованным алгоритмом, что нельзя сказать об итоговом результате.

Если же сравнивать с перебором то он получает следующие значения (рис. 11).



Рисунок 11 – Кол-во проходов при прямом переборе

**ВЫВОД**

В конечном итоге мы получили работающую программу, выполняющую все необходимые требования за фиксированное кол-во итерация. А также получили и развили навыки работы с различными алгоритмами.

**СПИСОК ИНФОРМАЦИОННЫХ ИСТОЧНИКОВ**

1. Страуструп Б. Программирование. Принципы и практика с использованием C++. 2-е изд., 2016.

2. Документация по языку С++ [Электронный ресурс]. URL: https://docs.microsoft.com/ruru/cpp/cpp/ (дата обращения 01.09.2021).

3. Курс: Структуры и алгоритмы обработки данных. Часть 2 [Электронный ресурс]. URL: https://online-edu.mirea.ru/course/view.php?id=4020 (дата обращения 01.09.2021).