Учреждение образования

«Белорусский государственный технологический университет»

**Отчет по лабораторной работе №1**

**Вспомогательные функции**

Выполнил:

Студент ФИТ 2-8-2

Данилецкий Максим

**Цель работы:** приобретение навыков составления и отладки программ с использованием пользовательских функций для замера продолжительности процесса вычисления.

В соответствии с заданием, были созданы и заполнены кодом 4 файла: 2 заголовочных, содержащих директивы include c нужными библиотеками и прототипами необходимых для выполнения функций, а также файл с реализацией функций iget, dget, start и основного проекта.

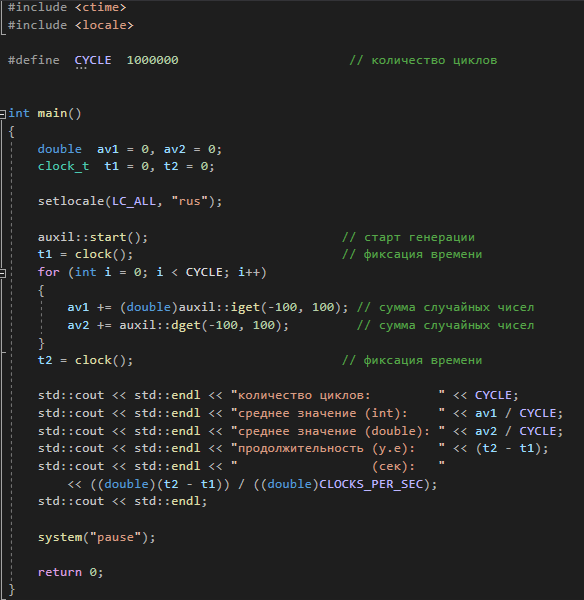


Рисунок 1 – Основной проект

С изменением параметра CYCLE при каждом запуске соответственно меняется время выполнения.

Вторым алгоритмом выберем подсчёт числа Фибоначчи, код на рисунке 2, где N – число, факториал которого считается.

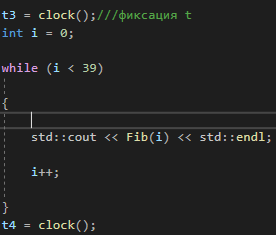


Рисунок 2 – подсчет факториала

На рисунке 2 заметно, что используется функция Fib.

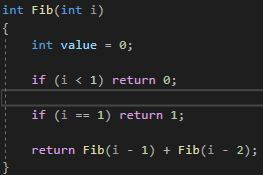


Рисунок 3 – функция Fib

График подсчета количества циклов представлен на рисунке 4.



Рисунок 4 – Графики скорости выполнения двух алгоритмов и таблицы результатов выполнения

Как мы можем видеть, время выполнения от кол-ва циклов зависит линейно. С точки зрения нотации Big O этот алгоритм является O(n).

График подсчета числа Фибоначчи представлен на рисунке 5.

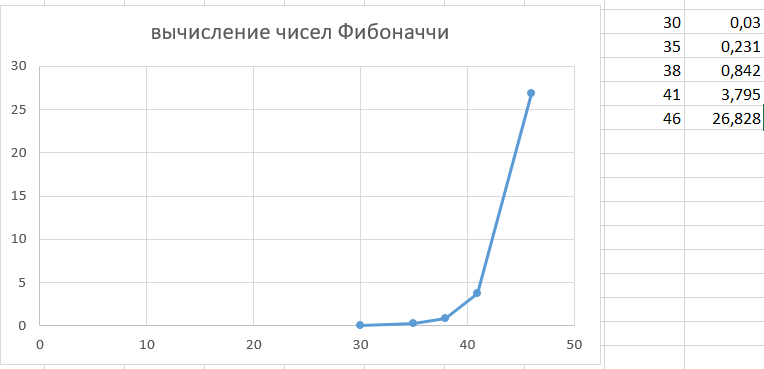


Рисунок 5 ­– График подсчета числа Фибоначчи

При вычислении числа Фибоначчи время выполнения зависит экспоненциально. С точки зрения Big O этот алгоритм является O(n^2). Причем разные алгоритмы требуют совершенно разное время на выполнение.

**Вывод:** В результате лабораторной работы приобретены навыки составления и отладки программ с использованием пользовательских функций для замера продолжительности процесса вычисления. Исследована зависимость времени выполнения расчёта суммы случайных чисел и подсчета числа Фибоначчи.

**Лабораторная работа №2**

**Комбинаторные алгоритмы решения оптимизационных задач**

**ЦЕЛЬ РАБОТЫ:** приобрести навыки разработки генераторов подмножеств, перестановок, сочетаний и размещений на С++; научиться применять разработанные генераторы для решения задач о рюкзаке (упрощенную, коммивояжера, об оптимальной загрузке судна и об оптимальной загрузке судна с центровкой.

**Задание 1**. Генератор подмножества заданного множества

В заголовочном файле Combi.h имеем структуру subset, описывающую подмножество (переменная для мощности множества, кол-ва элементов текущего подмножества, массив индексов текущего подмножества, битовую маску, конструктор и методы для дальнейшей работы с подмножеством), т.к. применение структуры позволяет создавать несколько экземпляров одного генератора и упростить его вызов в других программах. Файл Combi.cpp содержит полноценное описание структуры.

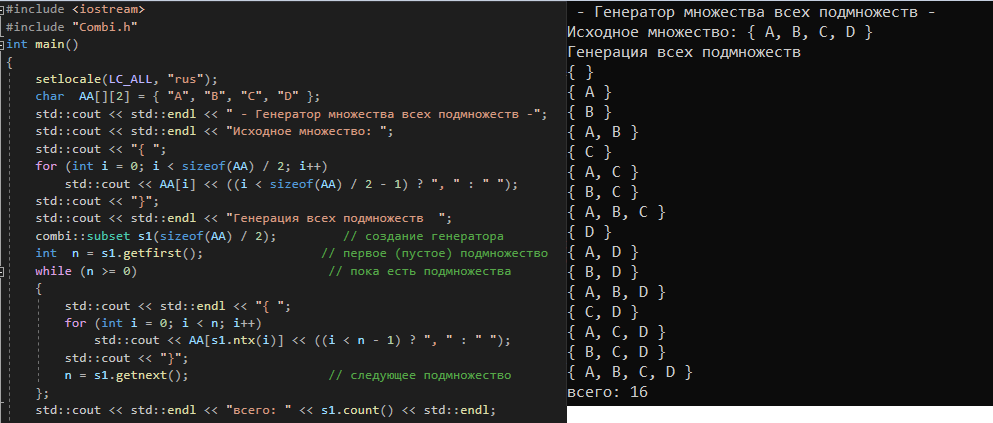


Рисунок 1 – Генерация подмножеств множества

Следует сказать, что алгоритм имеет сложность О(2^n) и поэтому реально может применяться для множеств с небольшой мощностью.

**Задание 2.** Генератор сочетаний.

Для начала создаём структуру xcombination, которая содержит переменную для кол-ва элементов исходного мн-ва, массив индексов текущего сочетания, а также функции для обхода множества. Реализация представлена на рисунке 2.

На основе генератора сочетаний решается задачи об оптимальной загрузке судна – необходимо выбрать такое кол-во контейнеров, чтобы их общий вес не превышал определенное значение, а доход от перевозки был бы максимально возможным. С помощью генератора сочетаний получаем всевозможные сочетания из кол-ва свободных мест по кол-ву контейнеров.

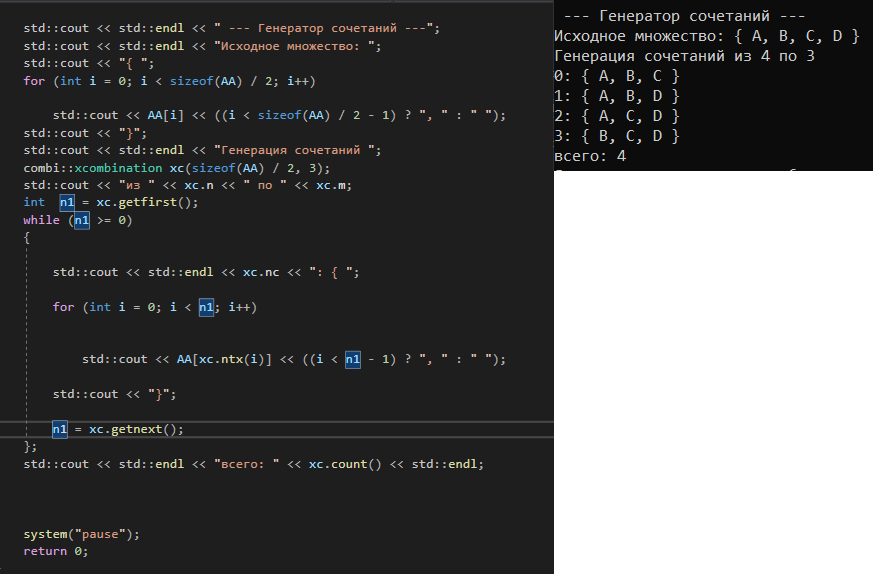


Рисунок 2 – Генератор сочетаний.

Такой алгоритм генерации сочетаний имеет сложность О(2^n), как и алгоритм генерации множества всех подмножеств.

**Задание 3.** Генератор перестановок.

Как и в прошлом задании, нам понадобится соответствующая структура данных и методы для работы с ней. В перестановках нам важен порядок, получим всевозможные перестановки.

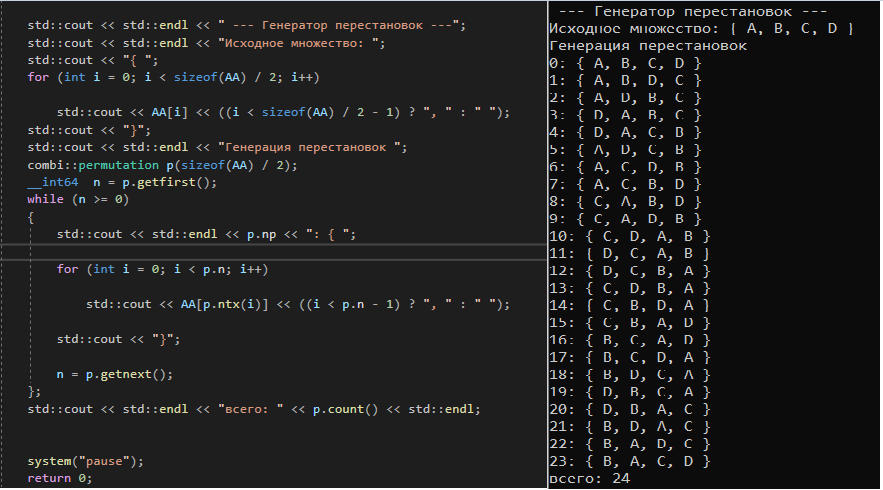


Рисунок 3 – Генератор перестановок

В алгоритме используется понятие ***мобильного элемента***. Элемент последовательности элементов множества называется мобильным, если соответствующая ему стрелка указывает на меньший соседний элемент. В первой перестановке все элементы кроме самого левого являются мобильными.

**Задание 4**. Генератор размещений

Аналогично прошлым заданиям, оп\*исываем структуру accommodation, которая содержит в себе переменные, индексирующие размещения и методы для работы с множеством. В размещениях учитывается порядок элементов.

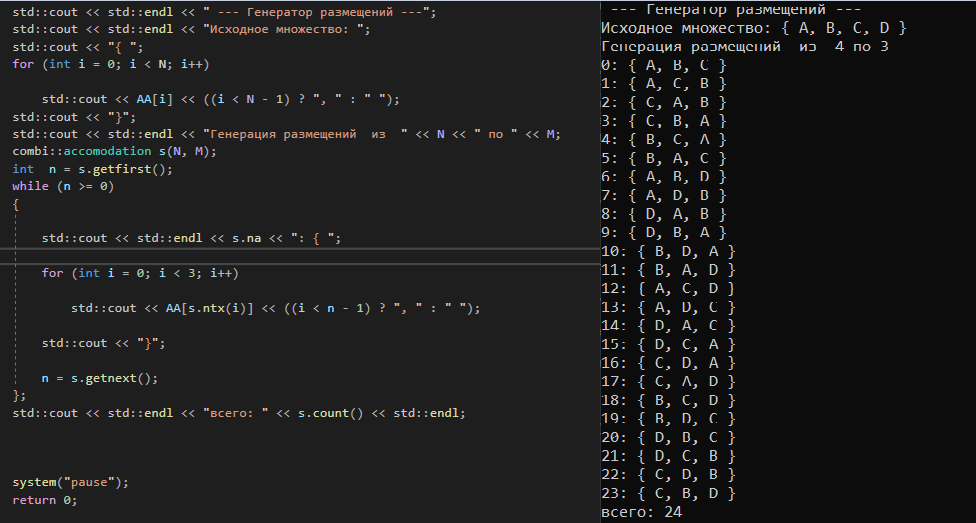
-

Рисунок 4 – Генератор размещений

Задание 5(вариант 3). Решить задачу об оптимальной загрузке судна (веса контейнеров сгенерировать случайным образом: ограничение по общему весу – 1500 кг., количество мест на судне для контейнеров – 5, количество контейнеров 25, веса контейнеров 100 – 900 кг., доход от перевозки 10 – 150 у.е.);

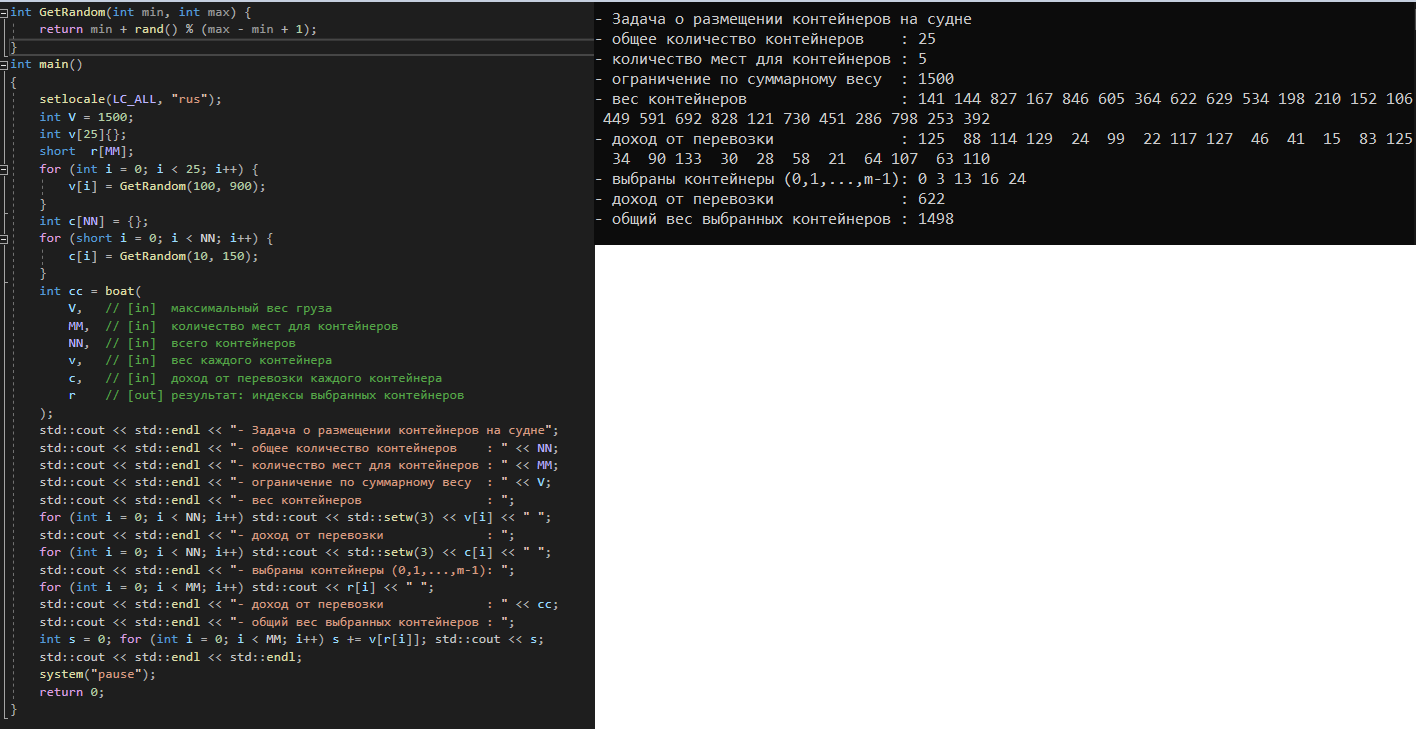


Рисунок 5. Задача об оптимальной загрузке судна

Функция **boat**последовательно генерирует все возможные сочетания по **m** контейнеров, вычисляет для каждого сочетания суммарный вес (функция **boatfnc::calcv**), для сочетаний с весом, не превышающим допустимое значение **V**,вычисляетдоход от перевозки этих контейнеров (**boatfnc::calcс**), фиксирует оптимальную комбинацию контейнеров (**boatfnc::copycomb**)и возвращает оптимальную доходность или нуль, если решения нет.

Задание 6.

**Лабораторная работа №3**

**МЕТОД ВЕТВЕЙ И ГРАНИЦ**

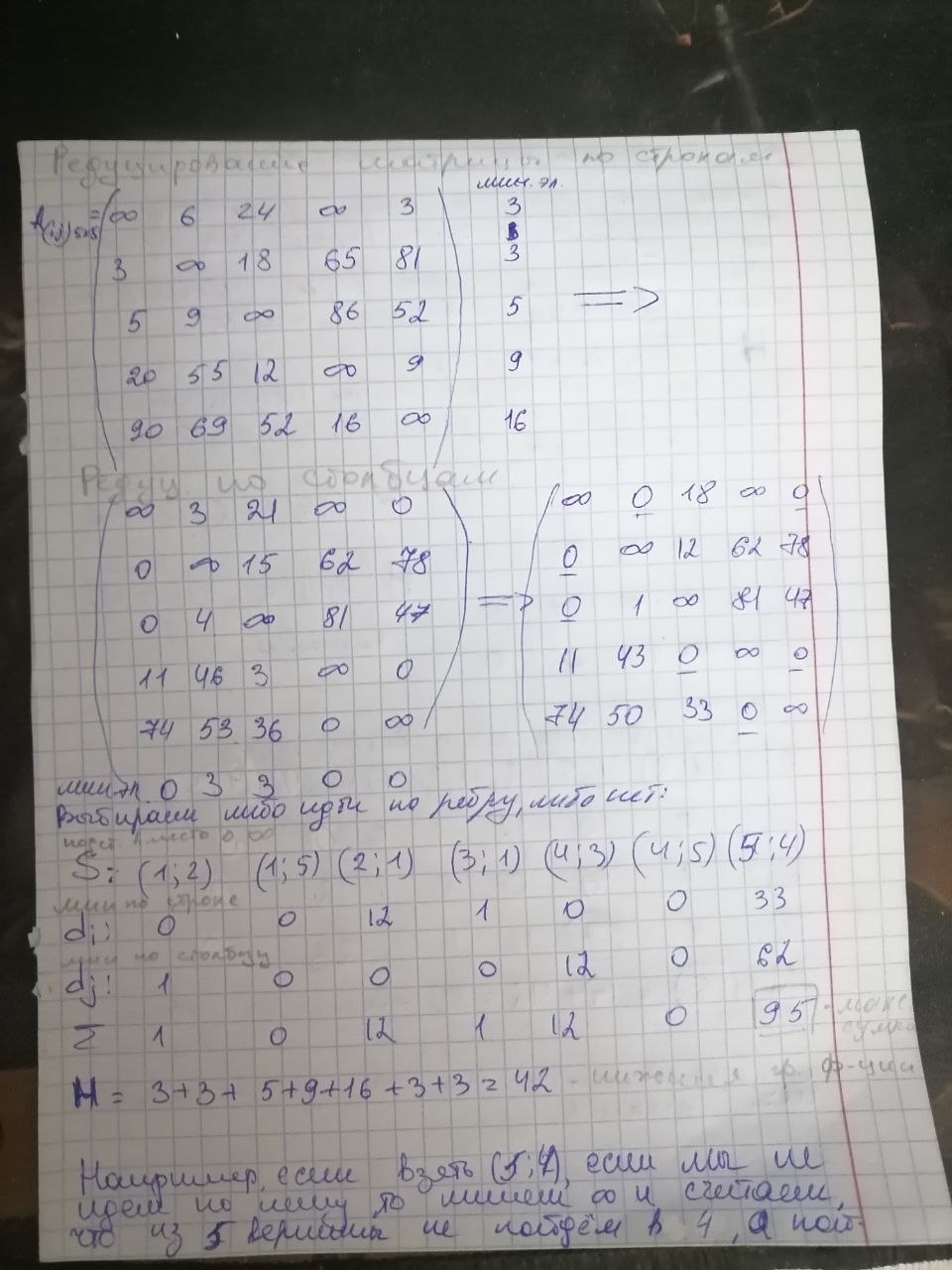
**ЦЕЛЬ РАБОТЫ:** освоить общие принципы решения задач методом ветвей и границ, решить задачу о коммивояжере данным методом, сравнить полученное решение задачи с комбинаторным методом перестановок.

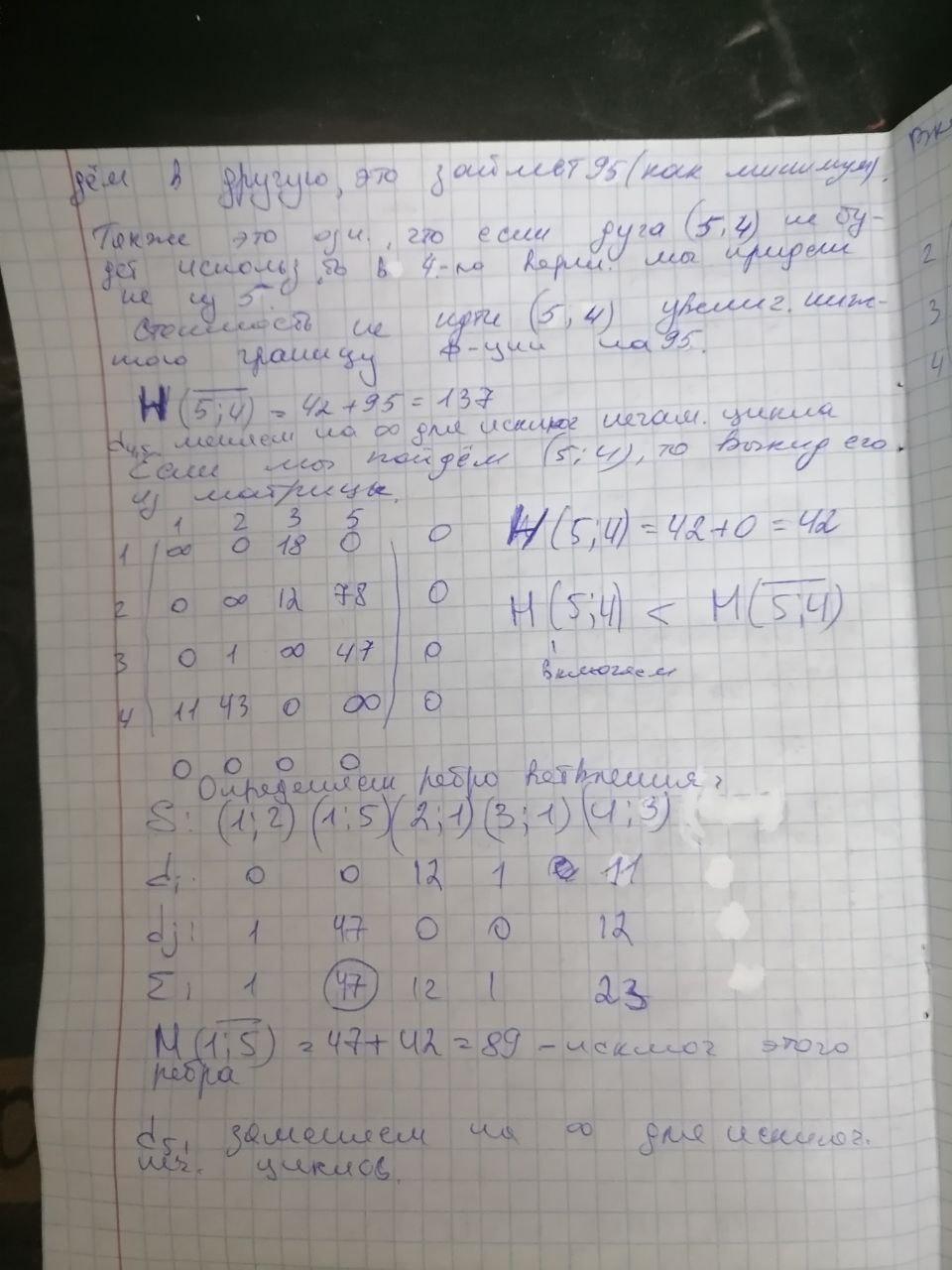
**Задание 1.** Сформулировать условие задачи коммивояжера с параметром.

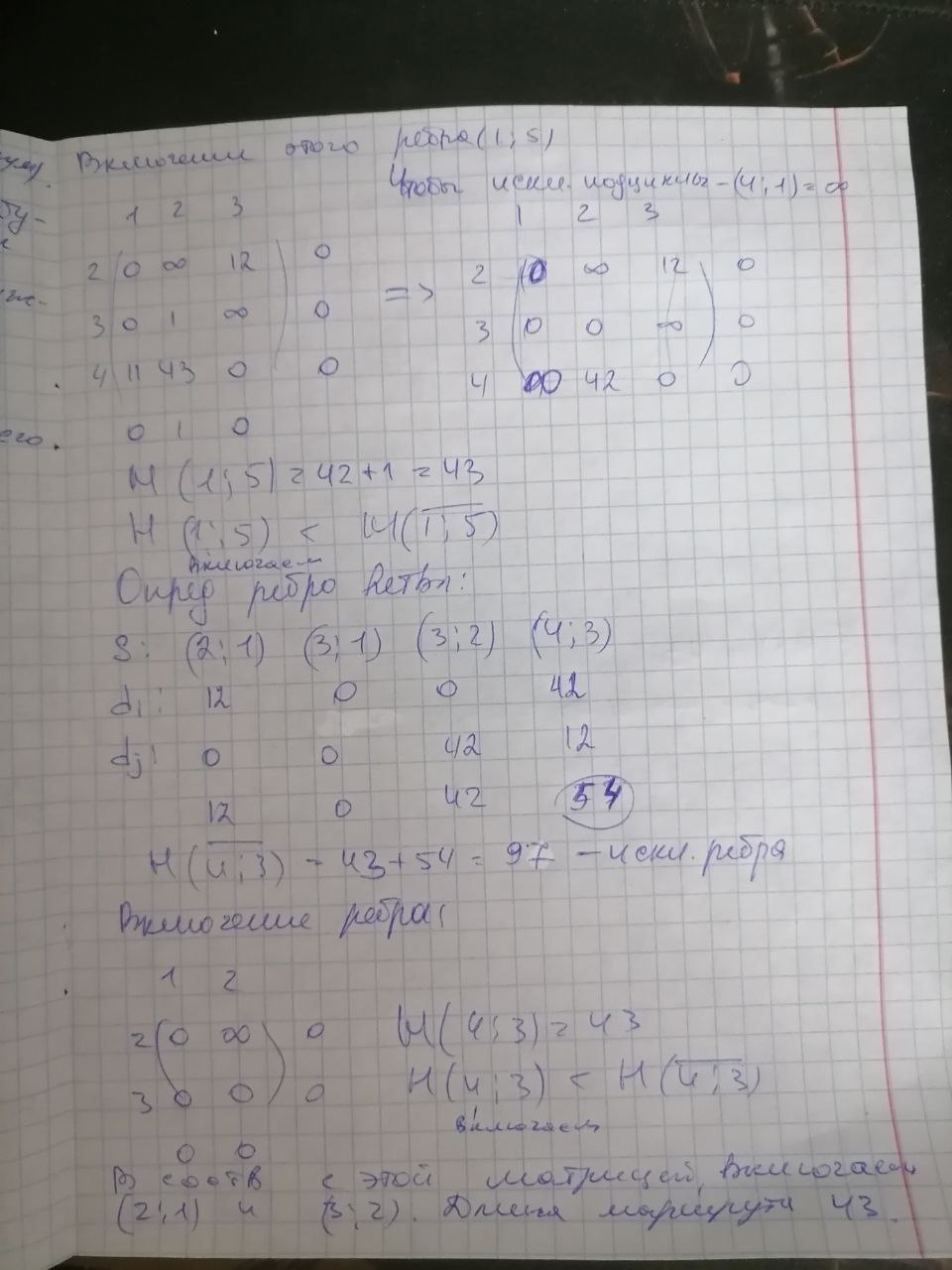
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Город** | **1** | **2** | **3** | **4** | **5** |
| **1** | INF | 6 | 24 | INF | 3 |
| **2** | 3 | INF | 18 | 65 | 81 |
| **3** | 5 | 9 | INF | 86 | 52 |
| **4** | 20 | 55 | 12 | INF | 9 |
| **5** | 90 | 69 | 52 | 16 | INF |

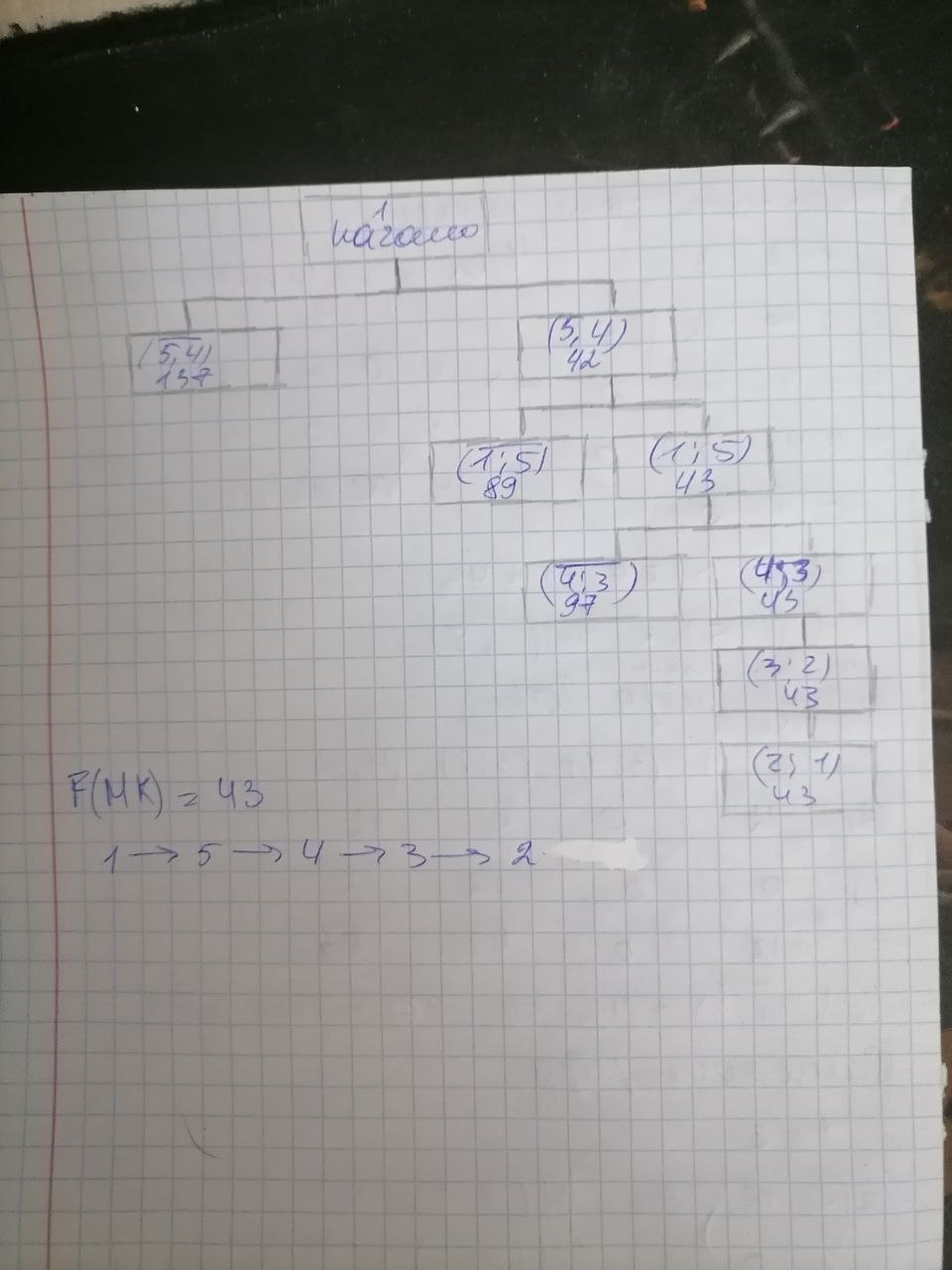
где *n* – номер варианта или номер по журналу (3);

**Задание 2.** Решить сформулированную задачу методом ветвей и границ.

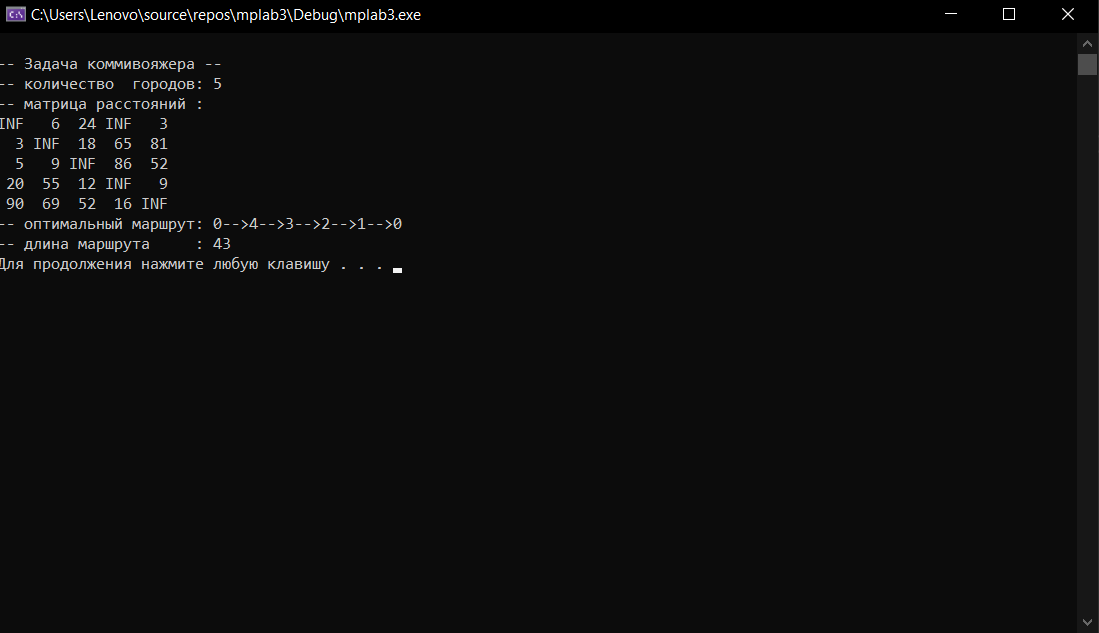








**Задание 3.** Проверить полученное решение при помощи генератора перестановок (см. лаб. 2, задание 5.1.) и включить копию экрана с решением в отчет.

****

**Вывод:** В этой лабораторной работе были освоены общие принципы решения задач методом ветвей и границ, решена задача о коммивояжере данным методом, а также сравнено полученное решение задачи с комбинаторным методом перестановок.