**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра МО ЭВМ**

отчет

**по лабораторной работе №1**

**по дисциплине «Построение и анализ алгоритмов»**

Тема: Поиск с возвратом

Рекурсивный бэктрекинг. Визуализация

|  |  |  |
| --- | --- | --- |
| Студент гр. 3388 |  | Березовский М.А. |
| Преподаватель |  | Жангиров Т.Р. |

Санкт-Петербург

2025

## Цель работы

Реализовать программу, которая визуализирует пошаговую работу алгоритма поиска с возвратом (backtracking) для задачи разбиения квадрата на под-квадраты.

## Задание

У Вовы много квадратных обрезков доски. Их стороны (размер) изменяются от 1 до N−1, и у него есть неограниченное число обрезков любого размера. Но ему очень хочется получить большую столешницу - квадрат размера N. Он может получить ее, собрав из уже имеющихся обрезков(квадратов).  
   Например, столешница размера 7×7 может быть построена из 9 обрезков.



   Внутри столешницы не должно быть пустот, обрезки не должны выходить за пределы столешницы и не должны перекрываться. Кроме того, Вова хочет использовать минимально возможное число обрезков.

**Входные данные**  
   Размер столешницы - одно целое число N (2≤N≤20).  
**Выходные данные**  
   Одно число *K*, задающее минимальное количество обрезков(квадратов), из которых можно построить  
столешницу(квадрат) заданного размера N*N*. Далее должны идти *K* строк, каждая из которых должна содержать три целых числа x, y и w, задающие координаты левого верхнего угла (1≤x,*y*≤*N*) и длину стороны соответствующего обрезка(квадрата).  
  
**﻿Пример входных данных**7 **Соответствующие выходные данные**9  
1 1 2  
1 3 2  
3 1 1  
4 1 1  
3 2 2  
5 1 3  
4 4 4  
1 5 3  
3 4 1

## Выполнение работы

**Структура проекта**

Программа разделена на три основных файла:

1. main.py – главный файл запуска программы.
2. solver.py – модуль с реализацией алгоритма (логика).
3. gui.py – модуль для графического интерфейса (визуализация).

Разделение сделано специально, чтобы было удобнее работать:

- вся логика поиска решения хранится отдельно от интерфейса

- интерфейс отвечает только за отображение, но не вмешивается в алгоритм

- главный файл остаётся коротким и служит только для запуска

**Как работает main.py**

Когда пользователь запускает python3 main.py, управление передаётся сюда. Программа проверяет корректность ввода (например, N должно быть ≥2). После этого создаётся объект визуализации, который автоматически создаёт окно, запускает алгоритм и ждёт действий пользователя.

**Как работает solver.py**

1. Алгоритм ищет свободную клетку
2. Пытается поставить туда квадрат разного размера
3. Если квадрат влезает, он размещается, и алгоритм углубляется дальше
4. Если тупик - происходит возврат (backtrack)
5. На каждом шаге вызывается yield, и GUI может показать текущее состояние

**Особенности:**

- Алгоритм реализован через генератор (yield). Это сделано для того, чтобы можно было возвращать не всё решение сразу, а отдельные шаги

- На каждом шаге алгоритм сообщает о своих действиях: например, что он поставил квадрат, сделал откат (backtrack) или нашёл лучшее решение. Таким образом, GUI получает «порции информации» и может их красиво отрисовывать.

**Как работает gui.py**

Создаются три основные зоны:

- Canvas (слева) – для рисования квадрата и под-квадратов

- Text (справа) – для вывода текстовых сообщений от алгоритма

- Frame с кнопками (снизу) – для управления (следующий шаг, авто, сброс)

Кнопки управления:

- «Следующий шаг» - запускает следующий yield из solver и отрисовывает его

- «Авто» – автоматически выполняет шаги с задержкой, пока алгоритм не закончится

- «Сброс» – очищает поле и запускает алгоритм заново.

Когда пользователь запускает программу, появляется окно.

- Сетка квадрата отрисована заранее

- При каждом шаге алгоритма закрашивается новый квадрат, а справа пишется сообщение

- Можно либо вручную щёлкать шаги, либо включить авто-режим

**Тестирование. Демонстрация граничных случаев алгоритма.**

Результаты тестирования представлены в табл. 1.

Таблица 1 – Результаты тестирования

|  |  |  |  |
| --- | --- | --- | --- |
| № п/п | Входные данные | Выходные данные | Комментарии |
|  | 2 |  | Верный вывод |
|  | 20 |  | Верный вывод |
|  | 7 |  | Верный вывод |
| 4. | 9 |  | Верный вывод |
| 5 | 16 |  | Верный вывод |