

**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»**

Студент гр. 7304

\_\_\_\_\_

Дементьев М.Е.

Преподаватель

\_\_\_\_\_

Филатов А.Ю.

Санкт-Петербург

2019

## Цель работы

Изучение алгоритма backtracking-a с помощью задачи на квадрирование квадрата размером  $N \times N$  квадратами размера от 1 до  $N-1$ .

## Описание алгоритма

Производится поиск пустой незакрашенной точки, от которой строится максимально большой квадрат. Если он заходит ровно (нет пробелов слева, снизу или справа), то рисование продолжается в цикле. Каждая точка, от которой был построен квадрат, заносится в временный ответ.

Если очередной построенный квадрат зашел неровно (есть хотя бы один пробел слева, снизу или справа), то от этого пробела вызывается рекурсивная функция с максимальной возможной длинной стороны квадрата уменьшенной на 1.

В конце сравнивается временный ответ с конечным, и, если первый меньше, то он становится конечным.

## Ход работы

- 1) Создание класса `quadrant`, который содержит в себе поле с длинной стороны квадрата и поле с его координатами( $x, y$ ). Координаты, в свою очередь, являются еще одним созданным классом: `coordinates`.
- 2) Создание функции `buildingASquare`, которая принимает площадку для построения квадрата (исходный большой квадрат), максимальное значение длинны квадрата, который необходимо построить, координаты для построения и цвет нового квадрата. Данная функция производит построение квадрата в указанных координатах и возвращает получившуюся длину стороны.
- 3) Создание функции `freeSpaceSearch`, которая принимает исходных квадрат и координаты. Она производит перебор всех позиций исходного квадрата и записывает в переданную ей переменную

координат – координаты первой встретившейся незанятой точки, вернув при этом true. Если свободное место не было найдено – возвращает false.

- 4) Написание функции `checkForGaps`, которая принимает исходный квадрат, координаты только что построенного нового квадрата и его длину. Она осуществляет проверку того, насколько ровно зашел новый квадрат. Если вокруг него не было найдено пустых клеток, возвращается true, иначе – false.
- 5) Создание основной рекурсивной функции `backtracking`, которая принимает исходный квадрат, максимальную длину новых квадратов, с помощью которых будет происходить квадрирование, цвет нового квадрата, массив с конечным ответом и массив с временным ответом, необходимый для сравнения и поиска наиболее оптимального варианта разбиения. Данная функция производит поиск наилучшего квадрирования квадрата заданного размера и вызывает в необходимые моменты все вышеуказанные функции.
- 6) Написание `main`-функции программы, которая считывает из консоли длину квадрата, квадрирование которого необходимо произвести, создает необходимые массивы, в которые в будущем, будут записываться ответы, вызывает основную рекурсивную функцию `backtracking` и производит вывод ответа на консоль с помощью функции `printAnswer`, которая, в свою очередь, принимает массив из ответов, проходит по нему и выводит все его элементы в консоль.

## Примеры работы программы

1) Входные данные: 4

Выходные:

```
4
1 1 2
3 1 2
1 3 2
3 3 2
```

2) Входные данные: 7

Выходные:

```
9
1 1 3
4 1 2
6 1 2
4 3 1
5 3 3
1 4 4
5 6 2
7 6 1
7 7 1
```

3) Входные данные: 15

Выходные:

```
6
1 1 5
6 1 10
1 6 5
1 11 5
6 11 5
11 11 5
```

## Вывод

В процессе выполнения лабораторной работы был реализован алгоритм поиска с возвратом (backtracking) и на его основе была решена задача квадрирования квадрата размером  $N \times N$  квадратами размера от 1 до  $N-1$ .

