**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. 6381 |  | Шевелева А.М. |
| Преподаватель |  | Филатов А.Ю. |

Санкт-Петербург

2018

**Цель работы**

Изучение алгоритма поиска с возвратом на основе задачи минимального заполнения квадрата размером NxN квадратами размера от 1 до N-1.

**Описание алгоритма**

После ввода числа N происходит проверка его на делимость на простые числа. Если делимость была обнаружена, то далее алгоритм разделяется на 2 части. Если первый простой делитель - 2, то срабатывает определенная схема, иначе определяется максимально возможное кол-во квадратов и максимальная сторона квадратов, а затем вызывается рекурсивная функция минимального заполнения.

При этом максимальная сторона берется N/2+1 и заполняется верхний левый угол, и два квадрата со стороной N/2 - верхний правый угол и нижний левый угол. Далее рекурсивно заполняются оставшиеся клетки.

**Описание функций**

void DeleteSquares (int \*\*solution, int quantity\_of\_squares, int \*\*box\_of\_squares)– Функция удаляет последний элемент в массиве решений.

void CreateSquares (int x, int y, int diagonal\_of\_squares, int \*\*solution, int quantity\_of\_squares, int \*\*box\_of\_squares)– Функция создания элемента.

void FirstSquares (int diagonal\_of\_squares, int diagonal\_of\_box, int \*\*solution, int \*\*box\_of\_squares) - Первое заполнение массива решений по оптимизации.

void SearchForPlaseForSquares (int &x, int &y, int diagonal\_of\_box,  int \*\*box\_of\_squares) - Поиск свободной клетки для заполнения.

bool CellTest (int x, int y, int diagonal\_of\_squares, int diagonal\_of\_box, int \*\*box\_of\_squares) - Проверка возможности вставки квадрата с переданной диагональю.

void FoundSolution (int \*\*solution, int \*\*answer, int &quantity\_of\_squares, int &min\_quantity\_of\_squares, int \*\*box\_of\_squares, int diagonal\_of\_box) - Сохранение решения в ответ и возврат на одну позицию.

**Тестирование**

Квадрат 7х7 - решение 9

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

Квадрат 6х6 - решение 4

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

Квадрат 5х5 - решение 8

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

Квадрат 9х9 - решение 6

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |

**Доказательства оптимизации**

1. Квадраты с четной стороной делятся минимально на 4 квадрата.

2. Воспользуемся алгоритмом полного перебора. Максимальная сторона, для которой мы смогли получить ответы - 29. Нарисуем график (сторона квадрата; минимальное кол-во квадратов).

Аппроксимируем данные точки в логарифмическую функцию.

В каждом из квадратов 3 - 29 были найдены квадраты со сторонами N/2+1 и N/2 (х2). Сделаем предположение, что и в последующих квадратах, с простыми сторонами будут содержаться квадраты такого вида. (Математический метод экстраполяции)

Оптимизируем наш алгоритм. Получим ответы для следующих квадратов - 31-61.

Полученные данные не выходят за рамки предположенной нами логарифмической функции, следовательно, можно сделать вывод, что такое предположение верно.

**Вывод**

В процессе выполнения лабораторной работы был реализован алгоритм поиска с возвратом и на его основе была решена задача минимального заполнения квадрата размером NxN.