

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

Ажель И.В.

Преподаватель

Филатов А.Ю.

Санкт-Петербург

2019

Цель работы:

Изучение алгоритма поиска с возвратом (backtracking).

Условия задания:

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).

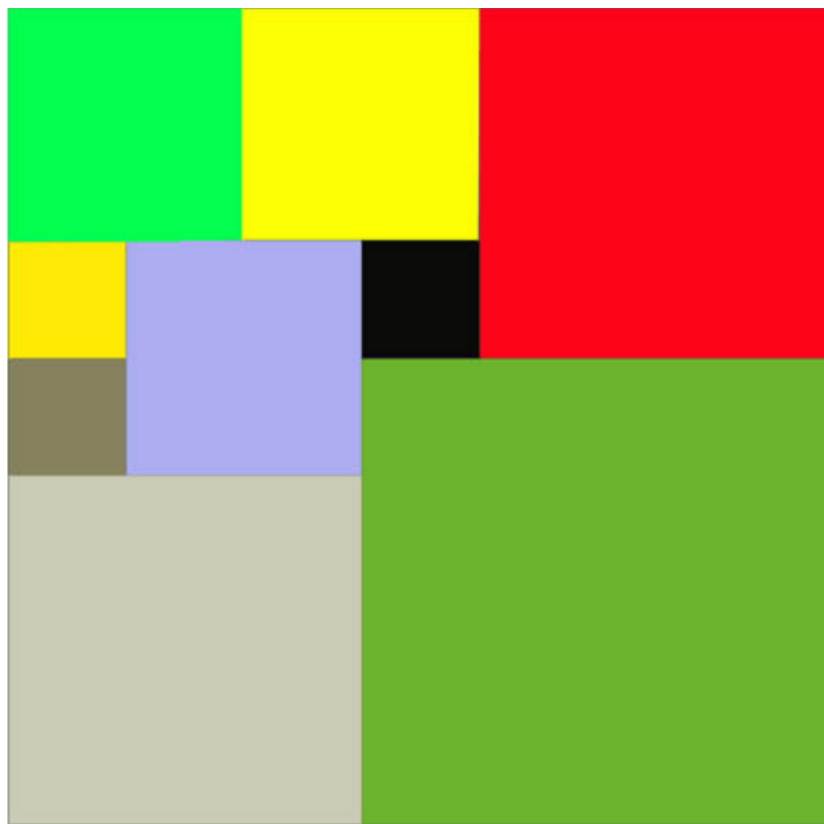


Рисунок 1. Пример разбиения квадрата со стороной 7.

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 ($2 \leq N \leq 20$).

Выходные данные:

Одно число K , задающее минимальное количество обрезков (квадратов), из которых можно построить столешницу (квадрат) заданного размера N . Далее должны идти K строк, каждая из которых должна содержать три целых числа x, y и w , задающие координаты левого верхнего угла ($1 \leq x, y \leq N$) 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

Описание функции

<pre>void backtracking(vector< vector<int> > field,int w,int N,int x,int y, int &min_size,int colour,int* result)</pre>	<p>Основная функция backtracking рекурсивна, суть её в следующем: Находится пустая точка, строится от неё максимально большой квадрат, если квадрат заходит ровно, то рисование продолжается в цикле, каждая точка, от которой происходит прорисовка, заносится в буферный ответ. На каждой итерации цикла снова происходит проверка на то, насколько ровно зашёл квадрат, если не ровно, то от этой точки вызывается рекурсивная функция с параметром максимального размера, уменьшенным на 1.</p>
---	---

Тестирование:

Ввод	Вывод
5	<div>Recursive iteration number:9</div> <div>1 1 2 3 3</div> <div>1 1 4 3 3</div> <div>5 5 5 6 6</div> <div>5 5 5 6 6</div> <div>5 5 5 7 8</div> <div>8</div> <div>1 1 2</div> <div>3 1 1</div> <div>4 1 2</div> <div>3 2 1</div> <div>1 3 3</div> <div>4 3 2</div> <div>4 5 1</div> <div>5 5 1</div>
9	<div>Recursive iteration number:7</div> <div>1 1 1 2 2 2 2 2 2</div> <div>1 1 1 2 2 2 2 2 2</div> <div>1 1 1 2 2 2 2 2 2</div> <div>3 3 3 2 2 2 2 2 2</div> <div>3 3 3 2 2 2 2 2 2</div> <div>3 3 3 2 2 2 2 2 2</div> <div>4 4 4 5 5 5 6 6 6</div> <div>4 4 4 5 5 5 6 6 6</div> <div>4 4 4 5 5 5 6 6 6</div> <div>6</div> <div>1 1 3</div> <div>4 1 6</div> <div>1 4 3</div> <div>1 7 3</div> <div>4 7 3</div> <div>7 7 3</div>

13

Recursive iteration number:12

```
1 1 1 1 1 1 2 2 2 2 2 2 2
1 1 1 1 1 1 2 2 2 2 2 2 2
1 1 1 1 1 1 2 2 2 2 2 2 2
1 1 1 1 1 1 2 2 2 2 2 2 2
1 1 1 1 1 1 2 2 2 2 2 2 2
1 1 1 1 1 1 2 2 2 2 2 2 2
3 3 3 3 4 4 2 2 2 2 2 2 2
3 3 3 3 4 4 5 6 6 6 6 6 6
3 3 3 3 7 7 7 6 6 6 6 6 6
3 3 3 3 7 7 7 6 6 6 6 6 6
8 8 8 9 7 7 7 6 6 6 6 6 6
8 8 8 10 10 11 11 6 6 6 6 6 6
8 8 8 10 10 11 11 6 6 6 6 6 6
```

11

1 1 6

7 1 7

1 7 4

5 7 2

7 8 1

8 8 6

5 9 3

1 11 3

4 11 1

4 12 2

6 12 2

Вывод:

В процессе выполнения лабораторной работы был изучен, разработан и реализован алгоритм поиска с возвратом для поиска ответа на задачу о минимальном заполнении квадрата размером $N \times N$ квадратами.