

Государственное образовательное учреждение высшего профессионального образования

"Московский государственный технический университет имени Н.Э.Баумана"

ФАКУЛЬТЕТ «Информатика и системы управления»

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

**Отчет по рубежному контролю №1
курса Анализ алгоритмов**

Студент: Денисенко А. А., группа ИУ7-646

Преподаватели: Волкова Л.Л., Строганов Ю.В.

Москва, 2020 г.

Содержание

1 Аналитическая часть	2
2 Конструкторская часть	3
2.1 Требования к программе	3
2.2 Разработка алгоритма	3
3 Технологическая часть	4
4 Экспериментальная часть	4

Введение

Задача данной работы - реализовать программу, выводящую массив размещений n единиц в строке нулей длиной m .

1 Аналитическая часть

Количество вариантов массивов, удовлетворяющих условию "длина массива равна m , количество единиц в массиве - n , а остальные элементы равны 0 можно вычислить по формуле сочетаний: $\frac{m!}{n!(m-n)!}$.

2 Конструкторская часть

2.1 Требования к программе

На вход программе подаются числа m и n . Необходимо вывести все варианты массивов, удовлетворяющих условию задачи.

2.2 Разработка алгоритма

Задача будет решаться рекурсивно. Глубина рекурсии равна $n + 1$, причем на самом глубоком уровне осуществляется только вывод полученного массива на экран. Решающая задачу функция принимает на вход число n , выполняющее роль счетчика глубины рекурсии, и начало обрабатываемой части массива $start$ (изначально 0). Алгоритм работы этой функции следующий:

- Вычесть 1 из счетчика глубины рекурсии
- Если счетчик больше 0, выполнить цикл со счетчиком $i = start..m$:
 - Приравнять элемент с индексом i к 1
 - Вызвать рабочую функцию, передать в нее текущее n и $start = i$
 - Приравнять элемент с индексом i к 0
- Если счетчик равен 0:
 - Вывести текущий массив на экран

3 Технологическая часть

Далее приведен листинг кода программы.

Листинг 1 Рекурсивная функция recursion

```
1 def recursion(n, start):
2     n -= 1
3     if n:
4         for i in range(start, m):
5             array[i] = 1
6             recursion(n, i+1)
7             array[i] = 0
8     else:
9         print(array)
```

Здесь n выполняет роль счетчика глубины рекурсии, start - начало изменяемой функцией части массива.

Листинг 2 Подготовка и вызов рекурсивной функции

```
1 m = input('Введите длину строки m: ')
2 n = input('Введите количество единиц n: ')
3
4 correct = True
5 try:
6     m = int(m)
7     n = int(n)
8 except:
9     correct = False
10
11 if n >= m:
12     correct = False
13
14 if correct:
15     array = [0]*m
16     recursion(n+1, 0)
17 else:
18     print('Некорректный ввод')
```

Здесь производится ввод пользователем m и n, проверка полученных значений на то, являются ли они целыми числами, и на то, является ли n большим числом чем m. Далее объявляется массив array, который будет меняться функцией recursion. В качестве аргумента start в recursion изначально подается 0.

4 Экспериментальная часть

Далее будут приведены результаты работы программы.

```
Введите длину строки m: 4
Введите количество единиц n: 5
Некорректный ввод
```

Рисунок 1 : Некорректный ввод

```
Введите длину строки m: 4
Введите количество единиц n: n
Некорректный ввод
```

Рисунок 2 : Некорректный ввод

```
Введите длину строки m: 5
Введите количество единиц n: 6
Некорректный ввод
```

Рисунок 3 : Некорректный ввод

```
Введите длину строки m: 5
Введите количество единиц n: 3
[1, 1, 1, 0, 0]
[1, 1, 0, 1, 0]
[1, 1, 0, 0, 1]
[1, 0, 1, 1, 0]
[1, 0, 1, 0, 1]
[1, 0, 0, 1, 1]
[0, 1, 1, 1, 0]
[0, 1, 1, 0, 1]
[0, 1, 0, 1, 1]
[0, 0, 1, 1, 1]
```

Рисунок 4 : Пример работы программы

```
Введите длину строки m: 3
Введите количество единиц n: 3
[1, 1, 1]
```

Рисунок 5 : Пример работы программы

```
Введите длину строки m: 6
Введите количество единиц n: 0
[0, 0, 0, 0, 0, 0]
```

Рисунок 6 : Пример работы программы