

Московский авиационный институт
(национальный исследовательский университет)

Факультет информационных технологий и прикладной
математики

Кафедра вычислительной математики и программирования

Лабораторная работа №7 по курсу «Дискретный анализ»

Студент: В. В. Косоголов
Преподаватель: А. А. Кухтичев
Группа: М8О-206Б
Дата:
Оценка:
Подпись:

Москва, 2020

Лабораторная работа №7

Задача: При помощи метода динамического программирования разработать алгоритм решения задачи, определяемой своим вариантом; оценить время выполнения алгоритма и объем затрачиваемой оперативной памяти. Перед выполнением задания необходимо обосновать применимость метода динамического программирования.

Вариант задания: Задано целое число n . Необходимо найти количество натуральных (без нуля) чисел, которые меньше n по значению и меньше n лексикографически (если сравнивать два числа как строки), а так же делятся на m без остатка.

1 Описание

Динамическое программирование - это способ решения сложных задач путём разбиения их на более простые подзадачи. Он применим к задачам, решение которых представимо как набор схожих с исходной подзадач меньшей сложности. В этом случае время вычислений, можно значительно сократить в сравнении с наивными методами.

Этапы построения алгоритма решения подзадач:

- 1) Описать структуру оптимального решения
- 2) Составить рекурсивное решение для нахождения оптимального решения.
- 3) Нахождение оптимального решения

2 Исходный код

Для решения данной задачи можно применить динамическое программирование, выделив следующие подзадачи: Для задачи с числами n , m , $len(n)$, где $len(n)$ - количество разрядов числа n подзадача будет состоять из количества чисел, удовлетворяющих условию и по длине равных $len(n)$:

$$T(n, m) = \frac{n}{m} - \frac{10^{len(n)-1}}{m}$$

```
#include <iostream>
#include <cmath>
long long length(long long num) {
    long long res = 0;
    while (num != 0) {
        num /= 10;
        res++;
    }
    return res;
}

int main() {
    long long n, m, res;
    std::cin >> n >> m;
    res = (n % m != 0) ? 0 : -1;
    while (n != 0) {
        if (length(n) != length(n - 1)) break;
        long long temp = std::pow(10, length(n) - 1);
        temp = (temp % m != 0) ? temp : temp-1;
        res += n/m - temp/m;
        n /= 10;
    }
    std::cout << (res < 0 ? 0 : res) << std::endl;
    return 0;
}
```

3 Консоль

```
...da_labs/lab_07 ./a.out  
42 3  
11
```

```
...da_labs/lab_07 ./a.out  
214892384928 15  
8510547033
```

```
...da_labs/lab_07 ./a.out  
5 2  
2
```

4 Выводы

Динамическое программирование является гибким методом построения алгоритмов. Всюду, где для решения задачи можно выделить подзадачи, динамическое программирование позволяет существенно ускорить работу программы.