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: Д. А. Тарпанов
Преподаватель: Н. С. Капралов
Группа: М8О-307Б-19
Дата:
Оценка:
Подпись:

Москва, 2021

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.

Разработать программу на языке C или C++, реализующую построенный алгоритм. Формат входных и выходных данных описан в варианте 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.

Формат входных данных

В первой строке задано $1 \leq n \leq 10^{18}$ и $1 \leq m \leq 10^5$.

Формат результата
Необходимо вывести количество искомых чисел.

1 Описание

Требуется реализовать алгоритм, решающий задачу с помощью методов динамического программирования. Разобьем задачу на более простые подзадачи: на каждом этапе будем рассматривать числа, состоящие из определенного количества разрядов, считать количество подходящих под условия задачи и добавлять их к результату.
Первый этап выполняется вне главного цикла, т.е. рассматриваются числа от 1 до последней цифры числа n, делящиеся на m без остатка. Далее, крассмотрению добавляется олин разряд, т.е. будут рассматриваться числа от 10 до двух последних цифр числа n и т.д. Для определения точных границ, то есть минимального и максимального рассматриваемых чисел, воспользуемся следующими суждениями: если $max\%m \neq 0$, то $max = max - (max\%m)$, если $min\%m \neq 0$, то $min = min + m - (min\%m)$. Тогда в промежутке между max и min будет лежать $(max/m) - (min/m) + 1$ подходящих нам чисел.
Временная сложность алгоритма составляет $O(|n| * |m|)$, т.е. зависят от длин входных чисел.

2 Исходный код

В файле main.cpp приводтся полное решение задачи.

Листинг main.cpp

```cpp
#include <iostream>

int main() {
    long long n, m;
    std::cin >> n >> m;

    if (n <= m){
        std::cout << 0 << '\n';
        return 0;
    }

    std::string str = std::to_string(n);
    long long dp = 0;

    for (int i = 1; i <= str[0] - '0'; ++i){
        if (i % m == 0){
            dp += 1;
        }
    }

    long long curMin = 10;
    std::string curStr;
    curStr += str[0];

    for (int i = 1; i < str.length(); ++i){
        curStr += str[i];
        long long min = curMin;
        curMin *= 10;
        long long max = std::stoll(curStr);
        if (max % m != 0){
            max = max - (max % m);
        }
        if (min % m != 0){
            min = min + m - (min % m);
        }
        if (max >= min){
            dp += (max / m) - (min / m) + 1;
        }
    }

    if (n % m == 0){
        std::cout << dp - 1 << '\n';
    } else {
```

```
44          std::cout << dp << '\n';
45      }
46      return 0;
47  }
```

3 Консоль

```
kng@kng-Legion-Y540-15IRH:~/CLionProjects/DA7/cmake-build-debug ./DA7
42 3
11
```

4 Тест производительности

Для изучения производетельности, сравним время работы программы на тестах разных размеров с $n = 10^8, 10^{10}, 10^{15}$ и $m = 12345$

```
1. $10^8$
/home/kng/CLionProjects/DA7/cmake-build-debug/DA7
1000000000
12345
5ms
Process finished with exit code 0

2.$10^10$
/home/kng/CLionProjects/DA7/cmake-build-debug/DA7
100000000000
12345
8ms
Process finished with exit code 0

2.$10^15$
/home/kng/CLionProjects/DA7/cmake-build-debug/DA7
10000000000000000
12345
22ms
Process finished with exit code 0
```

5 Выводы

Во время выполнения работы, я придумал алгоритм, решающий исходную задачу. Его сложно назвать динамическим программированием, так как зависимость текущего результата от прошлого выражается в простом суммировании. Я считаю, что это особенности конкретной задачи, и решить её другим, более похожим на ДП методом, наверное, нельзя.