

Московский авиационный институт
(национальный исследовательский университет)

Факультет информационных технологий и прикладной
математики

Кафедра вычислительной математики и программирования

Лабораторная работа №8 по курсу «Дискретный анализ»

Студент: Меджидли И. И. о
Преподаватель: Михайлова С. А
Группа: М8О-201Б-21
Дата: 26.02.2024
Оценка:
Подпись:

Москва, 2024

Лабораторная работа №8

Задача: На первой строке заданы два числа, N и $p > 1$, определяющие набор монет некоторой страны с номиналами p_0, p_1, \dots, p_{N-1} . Нужно определить наименьшее количество монет, которое можно использовать для того, чтобы разменять заданную на второй строке сумму денег $M \leq 2^{32} - 1$ и распечатать для каждого i -го номинала на i -ой строчке количество участвующих в размене монет. Кроме того, нужно обосновать почему жадный выбор неприменим в общем случае (когда номиналы могут быть любыми) и предложить алгоритм, работающий при любых входных данных.

1 Описание

Эту задачу можно решить с помощью метода жадных алгоритмов.

Жадный алгоритм — это алгоритм, заключающийся в принятии локально оптимальных решений на каждом этапе, допуская, что конечное решение также окажется оптимальным.

Этот код решает задачу о размене монет с использованием жадного алгоритма. Давайте разберемся, как он работает:

1. Сначала программа считывает три значения: N (количество монет), p (некоторый параметр) и $imedy$ (сумму, которую нужно разменять).
2. Создаются два массива: $resf$ и $ress$. $resf$ будет хранить степени p , а $ress$ — результаты размена.
3. Заполняется массив $resf$ значениями p^i для i от 1 до $N-1$.
4. Далее начинается жадный алгоритм:
 - 4.1. Инициализируется переменная $ch = N - 1$ (номер самой большой по номиналу монеты)
 - 4.2. Переменная $help$ получает значение $imedy$.
 - 4.3. Пока $imedy$ больше 0:
 - 4.4. Если $imedy$ (остаток на данный момент) больше или равняется $resf[ch]$ (номиналу самой большой монеты), то:
 - 4.4.1. Вычисляется остаток от деления $imedy$ на $resf[ch]$.
 - 4.4.2. В $ress[ch]$ записывается количество монет для размена: $ress[ch] = (help - imedy) / resf[ch]$;
 - 4.4.3. $help$ обновляется значением $imedy$.
 - 4.5. Иначе уменьшается значение ch .
5. В конце программа выводит результаты размена для каждой монеты (для каждого i -го номинала на i -ой строчке количество участвующих в размене монет.).

Важно: Жадный алгоритм не всегда дает оптимальное решение для задачи о размене монет, особенно если используется неканоническая монетная система. Например, если в задаче для размена 8 будут использоваться монеты номиналами 1, 4 и 5, то при решении жадным алгоритмом мы используем одну пятерку и три единицы. Но есть более оптимальное решение: две четверки.

Чтобы всегда получать оптимальное решение, можно решать такие задачи методом динамического программирования.

2 Исходный код

```
1 | #include <iostream>
2 | #include <cmath>
3 |
4 | using namespace std;
5 |
6 | int main()
7 | {
8 |     ios::sync_with_stdio(false); cin.tie(0);
9 |     long long N, p, imedy;
10 |    cin >> N >> p >> imedy;
11 |    long long resf[N];
12 |    long long ress[N] = {0};
13 |    resf[0] = 1;
14 |    for(long long i = 1; i < N; ++i){
15 |        resf[i] = pow(p, i);
16 |    }
17 |
18 |    long long ch = N - 1;
19 |    long long help = imedy;
20 |    while(imedy > 0){
21 |        if(imedy >= resf[ch]){
22 |            imedy %= resf[ch];
23 |            ress[ch] = (help - imedy) / resf[ch];
24 |            help = imedy;
25 |        }
26 |        else --ch;
27 |    }
28 |
29 |    for(auto &k : ress){
30 |        cout << k << "\n";
31 |    }
32 |    return 0;
33 | }
```

3 Тесты

Тестировать программу буду ручным способом.

```
imedzhidli@imedzhidli:~/Desktop/DA/LABA8$ ./lab8
3 5
71
1
4
2
imedzhidli@imedzhidli:~/Desktop/DA/LABA8$ ./lab8
3 3
148
1
1
16
imedzhidli@imedzhidli:~/Desktop/DA/LABA8$ ./lab8
10 3
1237843512
0
0
0
2
0
0
2
2
2
62888
imedzhidli@imedzhidli:~/Desktop/DA/LABA8$
```

Как можно заметить, все работает верно.

4 Выводы

При выполнении восьмой лабораторной работы по курсу «Дискретный анализ» я познакомился с жадными алгоритмами, смог решить "Задачу о размене монет". Также я доказал, что жадные алгоритмы не всегда дают оптимальные решения. Надеюсь, что полученные новые знания мне понадобятся в дальнейшем.