

Лабораторная работа №7 по курсу дискретного анализа: Динамическое программирование.

Выполнил студент группы М8О-308Б-22 Н. А. Ширяев

Условие

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 С или С++, реализующую построенный алгоритм.

Формат входных и выходных данных описан в варианте задания.

Имеется натуральное число n . За один ход с ним можно произвести следующие действия:

- вычесть единицу;
- разделить на два;
- разделить на три.

При этом стоимость каждой операции — текущее значение n . Стоимость преобразования — суммарная стоимость всех операций в преобразовании.

Вам необходимо с помощью последовательностей указанных операций преобразовать число n в единицу таким образом, чтобы стоимость преобразования была наименьшей. Делить можно только нацело.

Ограничения: $2 < n < 10^7$

Метод решения

Пусть после одной операции было получено число n' , тогда, очевидно, что $n' < n$. Тогда после одной операции наша задача, по сути, не изменилась: теперь требуется найти минимальное количество операций для числа n' , а не для числа n . Можно воспользоваться методом динамического программирования: заведем массив dp размера n , $dp[i]$ — минимальное количество операций требуемое для получения 1 из числа i . База динамического программирования $dp[1] = 0$, переходы из состояния i таковы: можно всегда перейти в состояние $i - 1$ за n , если i делится на 2, то можно перейти в состояние i за n , и, наконец, можно перейти в состояние i за n , если i делится на 3. Таким образом требуется выбрать лучшее из 3 возможных переходов решение. В итоге алгоритм имеет временную сложность $O(n)$ и пространственную сложность $O(n)$.

Описание программы

Ввиду простоты в реализации, лабораторная работа состоит из всего одного файла `main.cpp`.

Тест производительности

Сравнивать время работы алгоритма логично с алгоритмом полного перебора вариантов.

n	Полный перебор	Динамическое программирование
500	426 ms	3 ms
1000	33636 ms	3 ms
10000	>1 hour	6 ms

Как видно из таблицы, даже при малых значениях n полный перебор на порядки уступает динамическому программированию, это происходит из-за того, что переборный алгоритм имеет временную сложность сравнимую с экспоненциальной, что работает слишком долго при больших значениях n .

Выводы

В данной лабораторной работе я познакомился с методом динамического программирования. Научился применять данный важный метод для решения задач, которые на первый взгляд нельзя решить лучше, чем простым перебором.