

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

**"МИРЭА - Российский технологический университет"**

**РТУ МИРЭА**

Институт информационных технологий (ИТ)

Кафедра математического обеспечения и стандартизации информационных технологий (МОСИТ)

**ОТЧЕТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ №8.2**

**по дисциплине**

**«Структуры и алгоритмы обработки данных»**

Тема. ОТДЕЛЬНЫЕ ВОПРОСЫ АЛГОРИТМИЗАЦИИ

Реализация алгоритмов на основе сокращения числа

переборов

|  |  |
| --- | --- |
| Выполнил студент группы ИКБО-43-23 | Кощеев М. И. |
| Принял старший преподаватель | Рысин М.Л. |

Москва 2024

**СОДЕРЖАНИЕ**

[1 ЦЕЛЬ 3](#_Toc182746314)

[2 ХОД РАБОТЫ 4](#_Toc182746315)

[**4** **ВЫВОДЫ** 8](#_Toc182746316)

[**5** **ИСПОЛЬЗУЕМЫЙ ИСТОЧНИК** 9](#_Toc182746317)

# ЦЕЛЬ

Освоить приёмы реализации алгоритмов на основе сокращения числа

переборов

# 2 ХОД РАБОТЫ

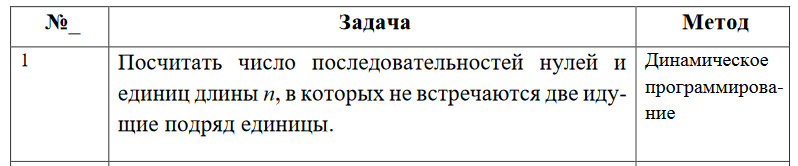
Разработать алгоритм решения задачи с применением метода, указанного

в варианте и реализовать программу.

Оценить количество переборов при решении задачи стратегией «в лоб» -

грубой силы. Сравнить с числом переборов при применении метода.

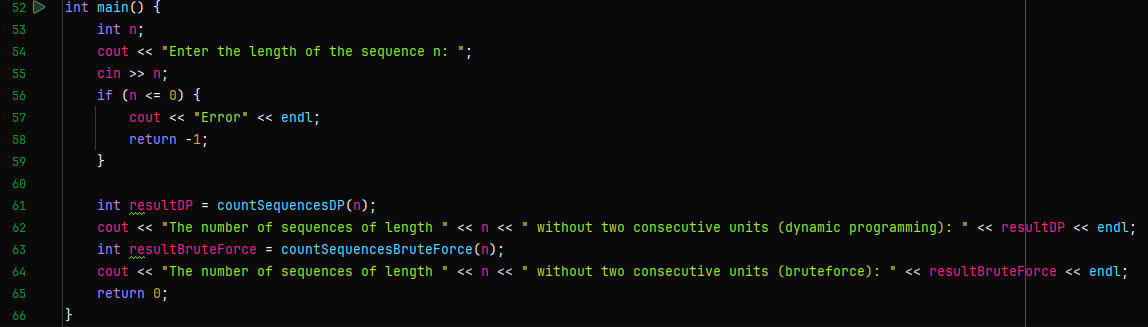
Вариант 1:



Алгоритм (main)

1. Ввод с клавиатуры длинны n
2. Вывод результата, используя динамическое программирование
3. Вывод результата, используя метод грубой силы (брутфорс)

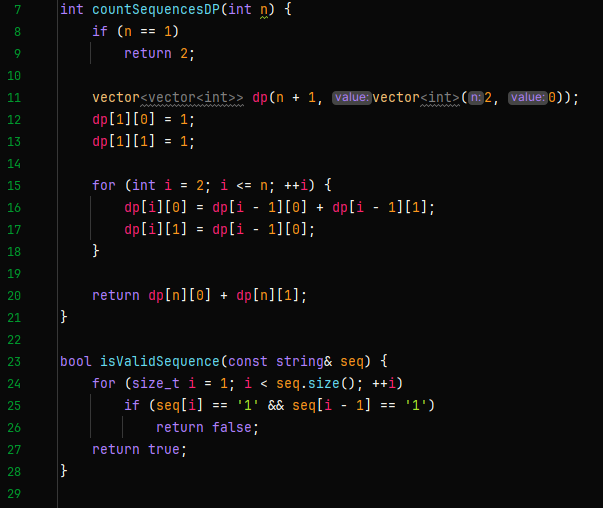
Код программы



Алгоритм (countSequencesDP):

1. Если n равен 1, то вывод 2
2. Создание вектора dp (количество последовательностей длинны) размером (n+1)\*2
3. Установка dp[1][0] = 1 (одна последовательность длины 1, заканчивающаяся на 0)
4. Установка dp[1][1] = 1 (одна последовательность длины 1, заканчивающаяся на 1)
5. Для каждого i от 2 до n:
   1. Вычисление количества последовательности длинны i, заканчивающихся на 1
   2. Вычисление количества последовательности длинны i, заканчивающихся на 0
6. Вывод общего количества последовательностей длинны n

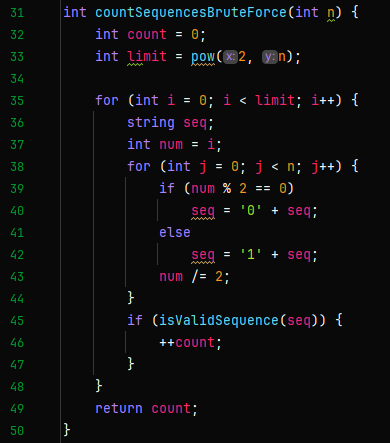
Код программы:



Алгоритм (countSequencesBruteForce)

1. Инициализация переменной count со значением 0
2. Инициализация переменной limit со значением 2n
3. Для каждого числа от 0 до n
   1. Инициализация строки seq
   2. Присваивание num значение i
   3. Для каждого числа в num проверяется, является ли текущая цифра в двоичной записи числом 0 или 1 (если 0, добавляем в начало seq 0, если 1, то 1)
4. Деление num на 2, чтобы перейти к следующей цифре в двоичной записи
5. Проверка, seq через isValidSequence. Если да, увеличиваем count на 1
6. Возврат значения count

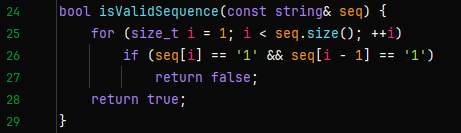
Код



Алгоритм (isValidSequence)

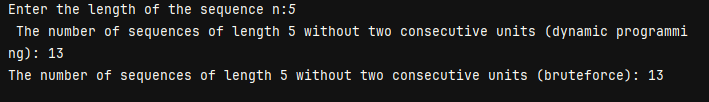
1. Проверка по 2 числа в строке
2. Если встречается 11, вывод false, если 11 не встречается – вывод true

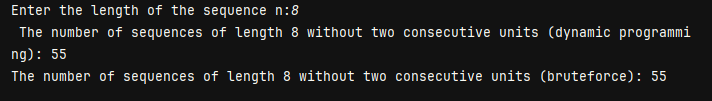
Код

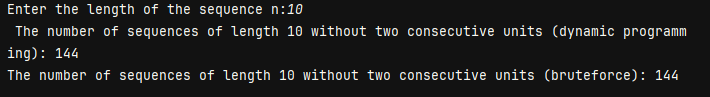


Тестирование

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Входные данные | Ожидаемый результат для ДП | Фактический результат для ДП | Ожидаемый результат для БФ | Фактический результат для БФ |
| 5 | 13 | 13 | 13 | 13 |
| 8 | 55 | 55 | 55 | 55 |
| 10 | 144 | 144 | 144 | 144 |







1. **ВЫВОДЫ**

В результате проделанной работы были освоены приёмы реализации алгоритмов на основе сокращения числа переборов (динамическое программирование и брутфорс)

1. **ИСПОЛЬЗУЕМЫЙ ИСТОЧНИК**

1. М.Л. РЫСИН, М.В. САРТАКОВ, М.Б. ТУМАНОВА Учебно-методическое

пособие СиАОД часть 2