

Федеральное государственное бюджетное образовательное учреждение
высшего образования

«Сибирский государственный университет телекоммуникаций и
информатики»
(СибГУТИ)

Институт информатики и вычислительной техники
Кафедра прикладной математики и кибернетики

Практическая работа №4
по дисциплине «Теория информации»
на тему «Определение параметров линейного кода»

Выполнил:

студент гр.ИП-014

Обухов А.И.

Проверила:

Старший преподаватель каф. ПМиК

Дементьева Кристина Игоревна

Новосибирск 2024 г.

Цель работы: Изучение свойств линейного корректирующего кода

Язык программирования: C, C++, C#, Python

Результат: программа, тестовые примеры, отчет.

Задание:

1. Порождающая матрица записана в текстовом файле. Файл имеет следующий формат: в первой строке через пробел записаны два натуральных числа n (количество строк матрицы) и m (количество столбцов), в следующих n строках записаны через пробел по m нулей и единиц. Файл можно генерировать случайно.

Пример файла

```
3 5
1 0 1 1 1
0 1 0 1 0
0 0 1 1 1
```

2. По заданной порождающей матрице определить характеристики линейного кода: размерность кода, количество кодовых слов, минимальное кодовое расстояние. Использовать 5 различных файлов.
3. Оформить отчет с результатами , загрузить отчет в электронную среду. По желанию в отчет можно включить описание программной реализации. **В отчет не нужно включать содержимое этого файла.**

Скриншоты работы программы

```
→ TI python3 lab4.py
Filename: output/matrix1.txt
Matrix Size: 2 x 5
Generator Matrix:
[[1 1 1 0 1]
 [1 1 1 1 1]]
Code Dimension: 2
Number of Code Words: 32
Minimum Code Distance: 1
```

```
Filename: output/matrix2.txt
Matrix Size: 4 x 8
Generator Matrix:
[[1 0 0 0 1 1 0 0]
 [1 0 0 0 1 0 0 0]
 [1 0 1 0 1 1 1 0]
 [0 1 0 0 1 1 1 1]]
Code Dimension: 4
Number of Code Words: 256
Minimum Code Distance: 1
```

```
Filename: output/matrix3.txt
Matrix Size: 3 x 6
Generator Matrix:
[[1 1 0 1 1 1]
 [0 1 1 1 1 1]
 [0 0 0 0 1 0]]
Code Dimension: 3
Number of Code Words: 64
Minimum Code Distance: 2
```

```
Filename: output/matrix4.txt
Matrix Size: 2 x 5
Generator Matrix:
[[1 1 0 1 1]
 [0 1 1 0 0]]
Code Dimension: 2
Number of Code Words: 32
Minimum Code Distance: 4
```

```
Filename: output/matrix5.txt
Matrix Size: 4 x 9
Generator Matrix:
```

```
Filename: output/matrix5.txt
Matrix Size: 4 x 9
Generator Matrix:
[[0 1 0 1 0 1 0 0 1]
 [0 1 0 1 0 0 0 0 1]
 [0 1 1 1 0 0 0 0 0]
 [1 1 0 0 0 1 1 0 0]]
Code Dimension: 4
Number of Code Words: 512
Minimum Code Distance: 1
```

Анализ результатов работы программы

1. Свойства линейного корректирующего кода:

- Ключевыми характеристиками кода являются размерность, количество кодовых слов и минимальное кодовое расстояние, которые влияют на эффективность исправления ошибок.
- Линейный кодирующий код позволяет обнаруживать и исправлять ошибки при передаче информации.

2. Зависимость характеристик кода от порождающей матрицы:

- Минимальное кодовое расстояние определяется внутренними свойствами матрицы, такими как количество единиц в каждой строке и их расположение.
- Размерность кода зависит от числа строк в порождающей матрице.
- Количество кодовых слов зависит от количества столбцов в порождающей матрице.

3. Влияние размерности и минимального кодового расстояния на надежность кода:

- Большая размерность кода приводит к возможности создания большего количества кодовых комбинаций, что обычно увеличивает его возможности для исправления ошибок.
- Большее минимальное кодовое расстояние обеспечивает лучшую способность кода обнаруживать и исправлять ошибки при передаче данных.

Листинг программы

```
import numpy as np
from typing import Tuple

def generate_generator_matrix(filename: str, n: int, m: int) -> None:
    generator_matrix = np.random.randint(0, 2, size=(n, m))
    with open(filename, 'w') as file:
        file.write(f"{n} {m}\n")
        for row in generator_matrix:
            file.write(' '.join(map(str, row)) + '\n')

def read_generator_matrix(filename: str) -> Tuple[np.ndarray, int, int]:
    with open(filename, 'r') as file:
        lines = file.readlines()
        n, m = map(int, lines[0].split())
        generator_matrix = np.array([list(map(int, line.split())) for line
in lines[1:]])
    return generator_matrix, n, m

def linear_code_characteristics(matrix: np.ndarray) -> Tuple[int, int,
int]:
    n, m = matrix.shape
    code_dimension = n
    code_words = 2 ** m
    min_distance = float("inf")
    for code in range(n):
        for i in range(code + 1, n):
            distance = np.sum(matrix[code] != matrix[i])
            min_distance = min(min_distance, distance)
    return code_dimension, code_words, min_distance

def main() -> None:
    num_of_matrices = 5
    for i in range(1, num_of_matrices + 1):
        filename = f"output/matrix{i}.txt"
        n = np.random.randint(2, 5) # ROWS
        m = np.random.randint(2, 10) # COLS
        generate_generator_matrix(filename, n, m)

    for i in range(1, num_of_matrices + 1):
        filename = f"output/matrix{i}.txt"
        generator_matrix, n, m = read_generator_matrix(filename)
        print(f"Filename: {filename}")
        print(f"Matrix Size: {n} x {m}")
        print("Generator Matrix:")
        print(generator_matrix)
        code_dimension, code_words, min_distance =
linear_code_characteristics(generator_matrix)
        print(f"Code Dimension: {code_dimension}")
        print(f"Number of Code Words: {code_words}")
```

```
        print(f"Minimum Code Distance: {min_distance}")
        print()

if __name__ == "__main__":
    main()
```