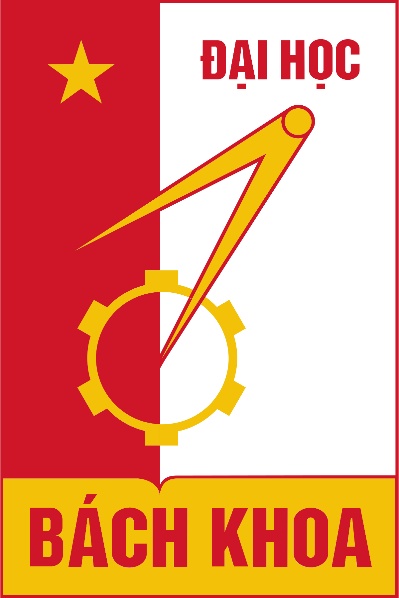
**TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI**

**VIỆN ĐIỆN TỬ - VIỄN THÔNG**



**BÁO CÁO BÀI TẬP LỚN**

**Môn: Cơ sở truyền tin**

**Đề tài: Mã thống kê tối ưu**

**GVHD: Nguyễn Hữu Phát**

**LỚP: TN – ĐIỆN TỬ TRUYỀN THÔNG K64**

**Nhóm thực hiện: Nhóm 1**

**Danh sách thành viên nhóm**

|  |  |
| --- | --- |
| Phạm Thanh Hải | 20191817 |
| Đào Đại Hiệp | 20193216 |
| Nguyễn Văn Thư | 20195662 |

**HÀ NỘI, NGÀY 13 THÁNG 1 NĂM 2022**

**MỤC LỤC**

Trang

[**I. CƠ SỞ LÝ THUYẾT 3**](#_Toc93010051)

[**1. Mã thống kê tối ưu 3**](#_Toc93010052)

[**1.1 Phương pháp mã hóa Shannon 3**](#_Toc93010053)

[**1.2 Phương pháp mã hóa Fano 3**](#_Toc93010054)

[**1.2.3 Phương pháp mã hóa Huffman 3**](#_Toc93010055)

[**II. GIỚI THIỆU ĐỀ TÀI 4**](#_Toc93010056)

[**III. CẤU TRÚC MÃ NGUỒN 4**](#_Toc93010057)

[**IV. CHẠY CHƯƠNG TRÌNH 6**](#_Toc93010058)

[**1. Chạy trực tiếp chương trình từ file main.exe 6**](#_Toc93010059)

[**2. Sử dụng API để chạy mô phỏng theo ý cá nhân 9**](#_Toc93010060)

[**3. Khảo sát 1 vài bài toán có dữ liệu lớn 9**](#_Toc93010061)

[**TÀI LIỆU THAM KHẢO 12**](#_Toc93010062)

# I. CƠ SỞ LÝ THUYẾT

## **1. Mã thống kê tối ưu**

- Là phép mã hóa mà kết quả là một bộ mã có chiều dài trung bình là nhỏ nhất trong tất cả các phép mã hóa có thể có trong nguồn

- Bộ mã của phép mã hóa tối ưu cho nguồn được gọi là mã hóa tối ưu

- Ba phép mã hóa là: Shannon, Fano, Huffman

Trong đề tài này sử dụng cơ số mã với a = 2 (mã hóa về dạng nhị phân)

Xét với một nguồn tin rời rạc X={x1, x2, …, xn} có kích cỡ nguồn tin là n với mỗi tin có xác suất p(x1, p(x2), …, p(xn)

Khi đó mỗi tin xi được mã hóa bằng một tổ hợp mã (từ mã) có độ dài là li, với mỗi giá trị tại 1 vị trí trong từ mã chỉ nhận 2 giá trị 0 hoặc 1.

Độ dài trung bình của bộ mã được xác định bằng công thức

### **1.1 Phương pháp mã hóa Shannon**

Các bước thực hiện:

B1: Sắp xếp nguồn tin theo thứ tự xác suất giảm dần

B2: Thay thế P(xi) = Pi theo công thức

B3: Đổi Pi từ thập phân sang chuỗi nhị phân

B4: Xác định độ dài từ mã theo công thức H(xi)<= li <= H(xi)+1

B5: Từ mã thu được của mỗi tin được lấy số kí tự sau dấu phẩy đúng bằng li

### **1.2 Phương pháp mã hóa Fano**

Các bước thực hiện:

B1: Sắp xếp nguồn tin theo thứ tự xác suất giảm dần hoặc tăng dần

B2: Chia nguồn tin thành 2 nhóm có tổng xác suất xấp xỉ nhau

B3: Gán cho 1 nhóm kí hiệu là ‘0’ nhóm còn lại là ‘1’

B4: Lặp lại B2 cho nhóm tin mới cho đến khi không thực hiện được nữa thì dừng lại

B5: Đọc từ mã theo chiều từ trái sang phải đó là từ mã thu được cho mỗi tin

### **1.2.3 Phương pháp mã hóa Huffman**

Các bước thực hiện:

B1: Sắp xếp nguồn tin theo thứ tự xác suất giảm dần hoặc tăng dần

B2: Chọn 2 tin có xác suất nhỏ nhất, gán 1 tin kí hiệu ‘0’ còn lại ‘1’

B3: Lặp lại B2 cho bộ tin mới cho đến khi chỉ còn 1 tin duy nhất

B4: Đọc từ mã theo chiều từ phải sang trái đó là từ mã thu được cho mỗi tin

# II. GIỚI THIỆU ĐỀ TÀI

Tên đề tài: Mô phỏng 3 loại mã thống kê tối ưu: Shannon, Huffman, Fano

Ngôn ngữ lập trình sử dụng: Python (Python 3)

Project gồm có 3 chức năng chính:

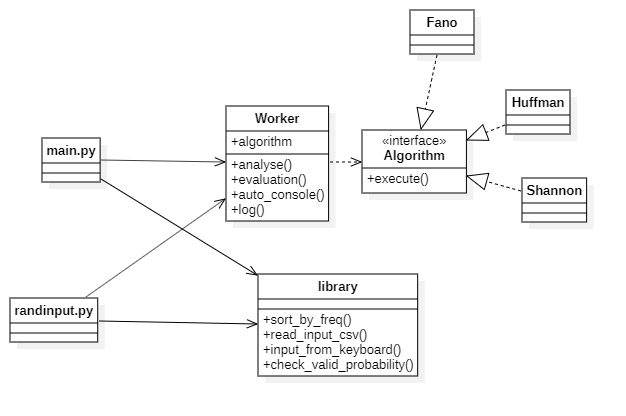
+ Tạo chương trình .exe có thể chạy trực tiếp trên hệ điều hành Windows, có thể đọc đầu vào từ bàn phím hoặc đọc đầu vào từ file .csv. Có kiểm tra đầu vào và in kết quả lên màn hình console và file result.csv

+ Tạo các API cho phép người dùng tuỳ biến chương trình, lấy ra các thông số cần thiết để phục vụ mục đích riêng

+ Áp dụng 3 thuật toán mã hoá thống kê tối ưu để khảo sát 1 vài bài toán có dữ liệu lớn

# III. CẤU TRÚC MÃ NGUỒN

Sơ đồ thiết kế lớp (class diagram):



Mã nguồn được tạo thành từ 3 file chính: Worker.py, library.py, Algorithm.py, được xây dựng theo mô hình Strategy Pattern

Worker.py có nhiệm vụ tính toán theo các phương án giải thuật đã cho, chứa các hàm xuất ra các thông số:

analyse(): xuất ra kết quả mã hoá, dưới dạng list of string

evaluation(), xuất ra lần lượt các giá trị: độ dài từ mã trung bình, entropy, hiệu suất

log(): in kết quả vảo file .csv

auto\_console(): tự động tính toán các thông số và in kết quả ra màn hình

Algorithm.py chứa 1 abstract class là Algorithm, được triển khai bằng 3 class tương ứng với 3 thuật toán: Fano, Huffman, Shannon. Hàm quan trọng trong các class này là execute(), trả về kết quả mã hoá dưới dạng list of string. Các hàm khác bổ trợ cho quá trình thực hiện thuật toán

library.py là thư viện chứa các hàm bổ trợ cho quá trình thực thi chương trình, gồm:

+ sort\_by\_freq(): sắp xếp nguồn tin theo thứ tự giảm dần của xác suất

+ read\_input\_csv() đọc dữ liệu từ file .csv, trả về lần lượt nguồn tin và xác suất

+ input\_from\_keyboard(): thực hiện việc nhập đầu vào từ bàn phím, có kiểm định dữ liệu hợp lệ

+ sum\_probability(): tổng xác suất

+ check\_valid\_probability(); kiểm tra chuỗi xác xuất có bằng 1 hay không

# IV. CHẠY CHƯƠNG TRÌNH

## **1. Chạy trực tiếp chương trình từ file main.exe**

+ file main.exe phải được đặt cùng cấp với file input.csv, chương trình sẽ nhắc nhở nếu tên file đầu vào không phải input.csv!

+ khuyến nghị sử dụng định dạng CSV (Comma delimited) để có kết quả chính xác nhất, định dạng CSV UTF-8 một vài trường hợp không cho kết quả tốt

+ file result.csv sẽ được sinh ra cùng cấp với 2 file trên, không nên xem kết quả này bằng phần mềm Excel, các từ mã sẽ bị mất các chữ số 0 ở đầu!

+ làm theo các hướng dẫn tiếp theo của chương trình

Kết quả khi chạy mode 1 với input cho sẵn

A picture containing graphical user interface

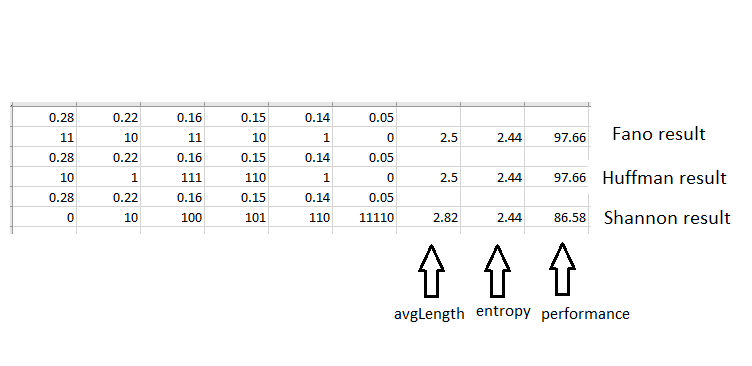
Description automatically generated

Kết quả khi chạy ở mode 2 với kiểm duyệt đầu vào:

A picture containing calendar

Description automatically generated

Đọc kết quả từ file result.csv



## **2. Sử dụng API để chạy mô phỏng theo ý cá nhân**

+ Các API được liệt kê tại mục I,

+ tạo ra một worker trước tiên:

worker = Worker()

+ lựa chọn thuật toán sẽ thực hiện trong đợt chạy này, có thể chọn Fano, Huffman, Shannon:

worker.algorithm = Fano()

worker.algorithm = Huffman()

worker.algorithm = Shannon()

+ lấy ra chuỗi từ mã: encoded = worker.analyse(data, freq)

+ lấy ra độ dài từ mã trung bình, entropy và hiệu suất:

avgLength, entropy, performance = worker.evaluation(result, freq)

## **3. Khảo sát 1 vài bài toán có dữ liệu lớn**

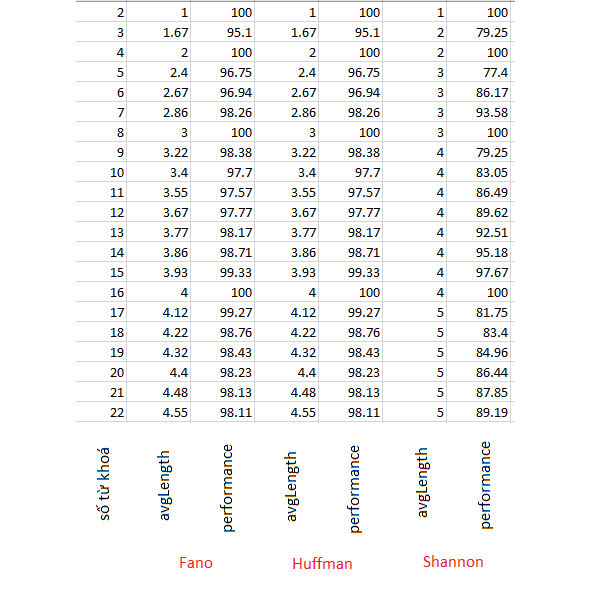
Nhóm đã khảo sát xem trong ba phương pháp mã hoá, phương pháp nào tốt nhất để mã hoá ngôn ngữ tiếng Anh, chuỗi xác suất lấy từ bảng xác suất các chữ cái trong từ điển tiếng Anh. Kết quả được log ra file /result\_dict/dict\_result:

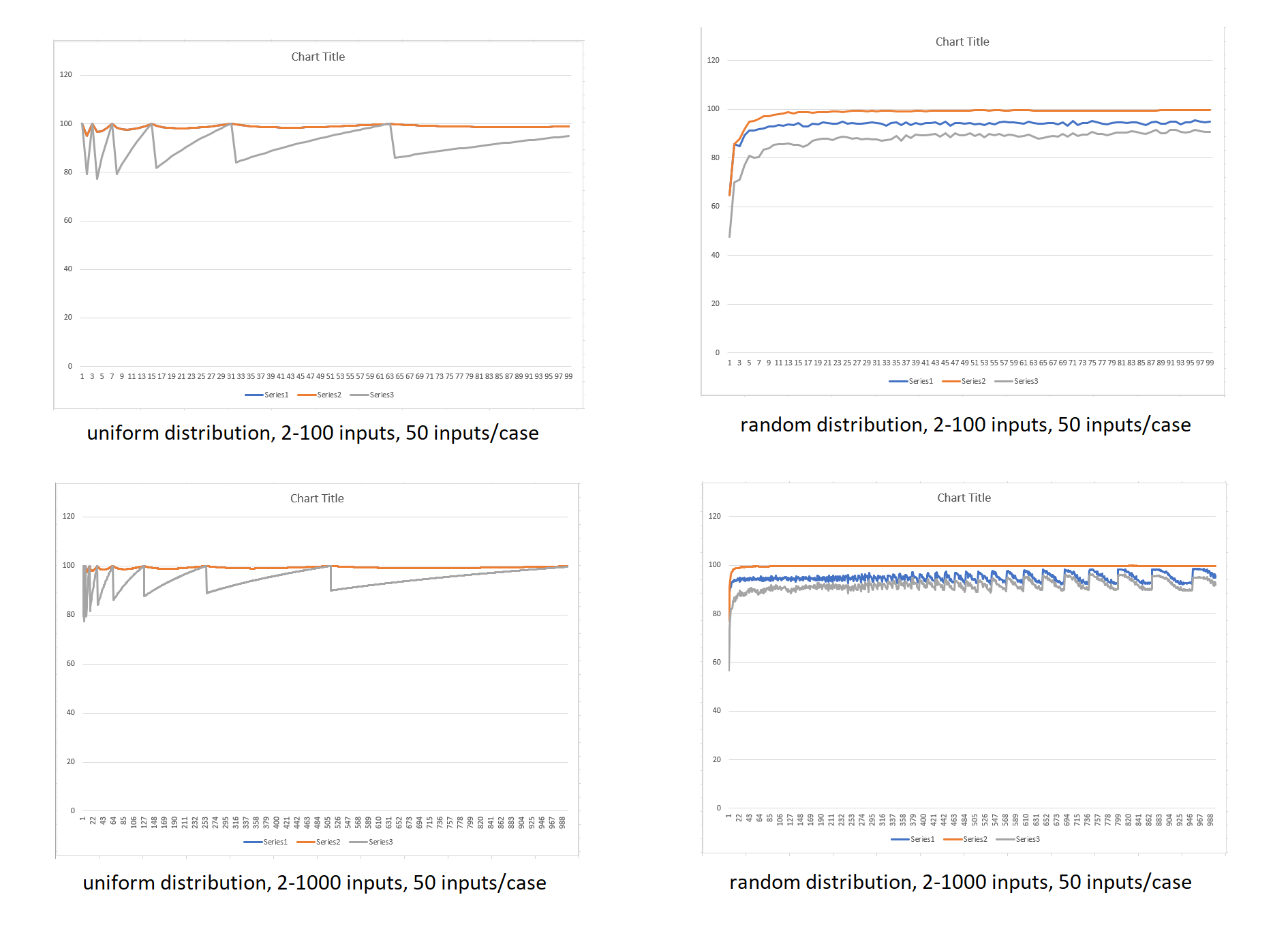
+ mã Fano cho hiệu suất 95.97%

+ mã Hufman cho hiệu suất 99.38%

+ mã Fano cho hiệu suất 90.52%

Nhóm thực hiện test với các nguồn tin có từ 2 đến 1000 phần tử đầu vào, với xác suất được phân bố theo: phân phối đều (uniform distribution) và 1 phân phối xác suất do Nhóm tự đưa vào (myrand), với mỗi 1 case có n phần tử với thuật myrand, Nhóm cho chạy 50 input khác nhau và tính giá trị trung bình hiệu suất, kết quả được log vào folder /distribution\_result:

Các kết quả được thể hiện bằng biểu đồ qua hình:



Đường màu xanh, cam, xám tương ứng với kết quả từ phương pháp Fano, Huffman, Shannon

+ với các nguồn tin có xác suất đều tuyệt đối, mã Fano và Huffman cho hiệu suất như nhau và tốt nhất, dự đoán, với phân phối gần đều, Huffman cho kết quả tốt nhất!

+ Với các nguồn tin có 1 phần tử có xác suất lớn vượt trội, mã Huffman cho kết quả tốt nhất, tiếp theo là Fano và Shannon. Hiệu suất trong trường hợp này của 2 phương pháp Fano và Shannon cho đồ thị khá thú vị như hình trên

# TÀI LIỆU THAM KHẢO

# [1] The frequency of the letters of the alphabet in English: <https://www3.nd.edu/~busiforc/handouts/cryptography/letterfrequencies.html>