## **ĐẠI HỌC QUỐC GIA TP.HCM**

## **ĐẠI HỌC KHOA HỌC TỰ NHIÊN**

## KHOA CÔNG NGHỆ THÔNG TIN





## BÁO CÁO ĐỒ ÁN

Môn: Cấu trúc dữ liệu và giải thuật

1. **Thông tin nhóm:**

|  |  |  |
| --- | --- | --- |
| **Họ và tên** | **MSSV** | **Gmail** |
| Võ Trọng Gia Vinh | 18120652 | vgia1055@gmail.com |
| Lê Thành Việt | 18120648 | lethanhviet7c@gmail.com |

**II. Mức độ hoàn thành:**

***1. Những câu đã làm được:***

* + Tìm tất cả công thức đa thức tối thiểu của hàm Bool.
  + Các phép toàn trên vector:
* Phép toán cộng hai vector.
* Phép toán nhân vector với 1 số alpha.
  + Các phép toán trên ma trận:
* Tìm định thức của ma trận.
* Nghịch đảo của ma trận.
* Tích hai ma trận.
* Tìm hạng của ma trận.
* Hệ phương trình tuyến tính.

***2. Những câu chưa làm được:*** 0

**II. Công việc của mỗi thành viên:**

1. **Lê Thành Việt:**

* Tìm công thức tối thiểu của hàm bool:
  + - Nhận input: string BoolExpression::inputBoolExpression()
    - Tách biểu thức thành những phần tử con và biến đổi sang dạng nhị phân: void BoolExpression::creatCell()
    - Xuất kết quả: void BoolExpression::printExpression()
* Các phép toán trên ma trận:
  + - Nhận input: void Matrix::inputMatrix().
    - Tìm định thức của ma trận:
    - double Matrix::determine(double\*\* m,int n).
    - int Matrix::determinantMatrix().
    - Tích hai ma trận: Matrix Matrix::multiplyMatrix(const Matrix& m).
    - Tìm hạng của ma trận: int Matrix::rankMatrix().
    - Xuất ma trận ra màn hình: void showMatrix();

1. **Võ Trọng Gia Vinh:**

* Tìm công thức tối thiểu của hàm bool:
* Xử lí biểu thức dưới dạng nhị phân và tìm công thức tối thiểu: void BoolExpression:: simplifyExpression().
* Tìm công thức rút gọn của biểu thức:

void BoolExpression::compactExpression().

* Các hàm xử lí vector:
* Nhân 2 vector với nhau:

Vector Vector:: operator +( const Vector &a)().

* Nhân vector với số alpha:

Vector Vector:: operator \*(double x)().

* Các phép toán trên ma trận:
* Giải phương trình tuyến tính:

void Matrix::systemLinerEquation().

* Tìm ma trận nghịch đảo:

void Matrix::inverseMatrix().