

**TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA CÔNG NGHỆ THÔNG TIN**



ĐỒ ÁN LẬP TRÌNH TÍNH TOÁN

ĐỊNH THỨC MA TRẬN

Người hướng dẫn: **Ths.Nguyễn Thị Minh Hỷ**

Sinh viên thực hiện:

Lê Minh Hoàng

Đinh Ngọc Trung

LỚP: 22T_Nhat2 NHÓM: 6

LỚP: 22T_Nhat2 NHÓM: 6

Đà Nẵng, ngày 23 tháng 6 năm 2023

MỤC LỤC

| | |
|---|----|
| MỤC LỤC | i |
| DANH MỤC HÌNH VẼ | ii |
| MỞ ĐẦU | i |
| 1. TỔNG QUAN ĐỀ TÀI | 1 |
| 2. CƠ SỞ LÝ THUYẾT | 1 |
| 2.1. Ý tưởng | 1 |
| 2.2. Cơ sở lý thuyết | 1 |
| 3. TỔ CHỨC CẤU TRÚC DỮ LIỆU VÀ THUẬT TOÁN | 1 |
| 3.1. Phát biểu bài toán | 1 |
| 3.2. Cấu trúc dữ liệu | 1 |
| 3.3. Thuật toán | 1 |
| 4. CHƯƠNG TRÌNH VÀ KẾT QUẢ | 3 |
| 4.1. Tổ chức chương trình | 3 |
| 4.2. Ngôn ngữ cài đặt | 3 |
| 4.3. Kết quả | 3 |
| 4.3.1. Giao diện chính của chương trình | 3 |
| 4.3.2. Kết quả thực thi của chương trình | 4 |
| 4.3.3. Nhận xét đánh giá | 6 |
| 5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN | 6 |
| 5.1. Kết luận | 6 |
| 5.2. Hướng phát triển | 6 |
| TÀI LIỆU THAM KHẢO | 7 |

DANH MỤC HÌNH VẼ

Hình 1: Giao diện chính của chương trình

Hình 2: Kết quả demo 1

Hình 3: Kết quả demo 2

Hình 4: Kết quả demo 3

MỞ ĐẦU

Trong phần này, cần trình bày về: Mục đích thực hiện đề tài, mục tiêu đề tài, phạm vi và đối tượng nghiên cứu, phương pháp nghiên cứu, cấu trúc của đồ môn học} Mục đích thực hiện đề tài: Thông qua quá trình thực hiện đề tài có thêm kiến thức, cũng như áp dụng được những kiến thức đã học để giải quyết vấn đề, bài toán một cách trơn tru tối ưu nhất có thể.

Mục tiêu đề tài: giải quyết được bài toán được đưa ra.

Phạm vi và đối tượng nghiên cứu: sử dụng bao quát những kiến thức đã học và tập trung chủ yếu tại các cấu trúc trong cấu trúc dữ liệu, đối tượng nghiên cứu chính là cấu trúc dữ liệu tập hợp rời rạc và ứng dụng của nó.

Phương pháp nghiên cứu: sử dụng các tài liệu được phổ cập trên trường cũng như tìm kiếm tài liệu từ Internet, đọc hiểu về cấu trúc dữ liệu tập hợp rời rạc, suy ra ý tưởng áp dụng kiến thức và hoàn thành đề bài.

Cấu trúc của đồ môn học: sau khi tìm hiểu các lý thuyết về về thuật toán và các lý thuyết liên qua lên ý tưởng và giải quyết vấn đề được đưa ra, kiểm tra và nâng cấp thuật toán.

1. TỔNG QUAN ĐỀ TÀI

Việc tính định thức bằng tay sẽ gây nhiều khó khăn và trở ngại như là với các ma trận lớn sẽ cần trải qua nhiều quá trình biến đổi phức tạp với nhiều thao tác dễ gây nhầm lẫn và sai sót trong quá trình tính toán, tốn nhiều thời gian cho quá trình học tập và công việc. Vì vậy, chương trình tính định thức giúp thực hiện các việc trên một cách nhanh chóng và tránh sai sót trong quá trình tính toán.

2. CƠ SỞ LÝ THUYẾT

2.1. Ý tưởng

Ý tưởng của đề tài này là tính định thức của một ma trận vuông bằng phương pháp Laplace. Sử dụng thuật toán đệ quy để khai triển Laplace.

2.2. Cơ sở lý thuyết

Phương pháp tính.

3. TỔ CHỨC CẤU TRÚC DỮ LIỆU VÀ THUẬT TOÁN

3.1. Phát biểu bài toán

Dữ liệu đầu vào:

- Ma trận vuông A có kích thước $n \times n$ (n là một số nguyên dương).
- Tên tệp tin đầu vào chứa ma trận A .
- Tên tệp tin đầu ra để ghi kết quả tính được.

Dữ liệu đầu ra:

- Kết quả tính được là định thức của ma trận A .
- Kết quả được in ra màn hình và ghi vào tệp tin đầu ra.

3.2. Cấu trúc dữ liệu

- Mảng 2 chiều $A[\text{MAX_SIZE}][\text{MAX_SIZE}]$: Đại diện cho ma trận vuông đầu vào.
- Biến nguyên n : Đại diện cho kích thước của ma trận vuông.
- Chuỗi ký tự $\text{input_file_name}[100]$: Đại diện cho tên tệp tin đầu vào.
- Chuỗi ký tự $\text{output_file_name}[100]$: Đại diện cho tên tệp tin đầu ra.
- Hàm $\text{determinant}()$: Đây là một hàm đệ quy để tính định thức của ma trận.

3.3. Thuật toán

Trình bày các thuật toán (phân tích độ phức tạp của các thuật toán:

Hàm đệ quy determinant():

- Nếu kích thước ma trận là 1, trả về phần tử duy nhất của ma trận.
- Nếu kích thước ma trận là 2, tính định thức theo công thức định sẵn.
- Nếu kích thước ma trận lớn hơn 2, tiến hành tính định thức theo phương pháp Laplace.
- Với mỗi cột của ma trận, tạo ma trận con bằng cách loại bỏ hàng và cột tương ứng.
- Gọi đệ quy để tính định thức của ma trận con.
- Tính định thức của ma trận gốc bằng cách cộng các định thức của ma trận con theo công thức Laplace.
- In ra màn hình quá trình tính toán định thức nếu level bằng 0.

- **Độ phức tạp thuật toán:** Độ phức tạp của đệ quy là $O(n!)$. Vì vậy, độ phức tạp của hai vòng lặp này là $O(n * n!)$.

-**Giải code:**

```
function determinant(A[MAX_SIZE][MAX_SIZE], n, level):
```

```
    det = 0
```

```
    submatrix[MAX_SIZE][MAX_SIZE]
```

```
    if n == 1:
```

```
        return A[0][0]
```

```
    else if n == 2:
```

```
        return A[0][0] * A[1][1] - A[0][1] * A[1][0]
```

```
    else:
```

```
        for k = 0 to n-1:
```

```
            for i = 1 to n-1:
```

```
                for j = 0 to n-1:
```

```
                    if j < k:
```

```
                        submatrix[i-1][j] = A[i][j]
```

```
                    else if j > k:
```

```
                        submatrix[i-1][j-1] = A[i][j]
```

```
sign = (k % 2 == 0) ? 1 : -1
sub_det = determinant(submatrix, n-1, level + 1)
det += A[0][k] * sign * sub_det

if level == 0:
    print("Buoc ", k+1, ": ", A[0][k], " * (", sign, ") * det(submatrix)")
    for i = 1 to n-1:
        for j = 0 to n-2:
            print(submatrix[i-1][j], " ")
        print()
    print()

return det
```

4. CHƯƠNG TRÌNH VÀ KẾT QUẢ

4.1. Tổ chức chương trình

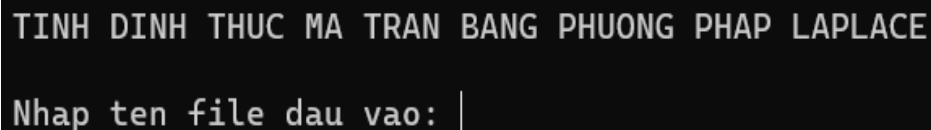
Hàm determinant để tính định thức và hiển thị các bước tính. Hàm printMatrix để in ma trận cần tính. Hàm main để nhập dữ liệu, thực thi các hàm trên và xuất dữ liệu.

4.2. Ngôn ngữ cài đặt :

Ngôn ngữ lập trình C

4.3. Kết quả

4.3.1. Giao diện chính của chương trình



```
TINH DINH THUC MA TRAN BANG PHUONG PHAP LAPLACE
Nhap ten file dau vao: |
```

4.3.2. Kết quả thực thi của chương trình

Mô tả kết quả thực hiện chương trình

DEMO 1:

```
TINH DINH THUC MA TRAN BANG PHUONG PHAP LAPLACE

Nhap ten file dau vao: dt1.inp
Ma tran can tinh:
  5.60   4.00   6.00  -4.70
  7.00   5.00   9.00   4.50
 23.00   4.00   3.00  -9.00
  4.00  -7.00   4.00  -1.00

Buoc 1: 5.60 * (1.00) * det(submatrix)
  5.00   9.00   4.50
  4.00   3.00  -9.00
 -7.00   4.00  -1.00

Buoc 2: 4.00 * (-1.00) * det(submatrix)
  7.00   9.00   4.50
 23.00   3.00  -9.00
  4.00   4.00  -1.00

Buoc 3: 6.00 * (1.00) * det(submatrix)
  7.00   5.00   4.50
 23.00   4.00  -9.00
  4.00  -7.00  -1.00

Buoc 4: -4.70 * (-1.00) * det(submatrix)
  7.00   5.00   9.00
 23.00   4.00   3.00
  4.00  -7.00   4.00

Nhap ten file dau ra: dt1_out
Dinh thuc ma tran can tinh = -12795.6000
Ket qua duoc luu vao file dt1_out

-----
Process exited after 125.6 seconds with return value 0
Press any key to continue . . . |
```

DEMO 2:

TINH DINH THUC MA TRAN BANG PHUONG PHAP LAPLACE

Nhap ten file dau vao: dt2.txt

Ma tran can tinh:

| | | | | |
|-------|-------|-------|-------|-------|
| 1.80 | 4.00 | 6.00 | -4.70 | 3.10 |
| 7.00 | 5.00 | 9.00 | 4.50 | 18.00 |
| 23.00 | 4.00 | 3.00 | -9.00 | 5.30 |
| 4.00 | -7.00 | 4.00 | -1.00 | 6.20 |
| 17.00 | 9.30 | 10.20 | 22.20 | 8.00 |

Buoc 1: $1.80 * (1.00) * \det(\text{submatrix})$

| | | | |
|-------|-------|-------|-------|
| 5.00 | 9.00 | 4.50 | 18.00 |
| 4.00 | 3.00 | -9.00 | 5.30 |
| -7.00 | 4.00 | -1.00 | 6.20 |
| 9.30 | 10.20 | 22.20 | 8.00 |

Buoc 2: $4.00 * (-1.00) * \det(\text{submatrix})$

| | | | |
|-------|-------|-------|-------|
| 7.00 | 9.00 | 4.50 | 18.00 |
| 23.00 | 3.00 | -9.00 | 5.30 |
| 4.00 | 4.00 | -1.00 | 6.20 |
| 17.00 | 10.20 | 22.20 | 8.00 |

Buoc 3: $6.00 * (1.00) * \det(\text{submatrix})$

| | | | |
|-------|-------|-------|-------|
| 7.00 | 5.00 | 4.50 | 18.00 |
| 23.00 | 4.00 | -9.00 | 5.30 |
| 4.00 | -7.00 | -1.00 | 6.20 |
| 17.00 | 9.30 | 22.20 | 8.00 |

Buoc 4: $-4.70 * (-1.00) * \det(\text{submatrix})$

| | | | |
|-------|-------|-------|-------|
| 7.00 | 5.00 | 9.00 | 18.00 |
| 23.00 | 4.00 | 3.00 | 5.30 |
| 4.00 | -7.00 | 4.00 | 6.20 |
| 17.00 | 9.30 | 10.20 | 8.00 |

Buoc 5: $3.10 * (1.00) * \det(\text{submatrix})$

| | | | |
|-------|-------|-------|-------|
| 7.00 | 5.00 | 9.00 | 4.50 |
| 23.00 | 4.00 | 3.00 | -9.00 |
| 4.00 | -7.00 | 4.00 | -1.00 |
| 17.00 | 9.30 | 10.20 | 22.20 |

Nhap ten file dau ra: dt1

Dinh thuc ma tran can tinh = 540134.3500

Ket qua duoc luu vao file dt1

DEMO 3:

TINH DINH THUC MA TRAN BANG PHUONG PHAP LAPLACE

Nhap ten file dau vao: dt3.txt

Ma tran can tinh:

| | |
|------|------|
| 1.00 | 2.00 |
| 2.00 | 4.00 |

Nhap ten file dau ra: dt1

Dinh thuc ma tran can tinh = 0.0000

Ket qua duoc luu vao file dt1

4.3.3. Nhận xét đánh giá

Với các trường hợp $n = 1, 2$.

- Hàm trả về định thức của ma trận ngay lập tức mà không cần thực hiện bất kỳ vòng lặp hay đệ quy nào. Vì vậy, độ phức tạp là $O(1)$.

Với các trường hợp với $n > 2$

- Hàm sử dụng hai vòng lặp for để lặp qua từng cột k của ma trận và tính định thức dựa trên đệ quy. Trong mỗi vòng lặp, hàm cũng tính ma trận con submatrix bằng cách loại bỏ dòng đầu tiên và cột k khỏi ma trận A . Sau đó, hàm gọi đệ quy để tính định thức của submatrix với kích thước $n-1$. Độ phức tạp của đệ quy là $O(n!)$. Vì vậy, độ phức tạp của hai vòng lặp này là $O(n * n!)$.

5. KẾT LUẬN

5.1. Kết quả đạt được

Kết quả thu được sau quá trình nghiên cứu là chương trình tính định thức ma trận phục vụ cho quá trình tính toán định thức ma trận trong học phần Đại số tuyến tính.

5.2. Hạn chế

Giao diện chương trình hiển thị vẫn chưa hoàn chỉnh. Không chạy trên các ngôn ngữ lập trình cao hơn như Java, Ruby, ...

5.3. Hướng phát triển

Thiết kế giao diện cho sản phẩm.

Áp dụng thuật toán vào các công cụ tính toán như máy tính bỏ túi để phục vụ trong các quá trình tính toán đo đạc phục vụ công việc học tập và làm việc.

TÀI LIỆU THAM KHẢO

[1] Độ Phức Tạp Của Thuật Toán Và Lựa Chọn Cách Giải Thuật :

<https://codelearn.io/sharing/do-phuc-tap-cua-thuat-toan-va-lua-chon-cach-giaithuat#:~:text=Tuy%20nhi%C3%AAAn%20v%E1%BB%9Bi%201%20s%E1%BB%91%20gi%E1%BA%A3i%20thu%E1%BA%ADt%20s%E1%BA%AFp,ch%E1%BA%A1y%20c%C3%B3%20th%E1%BB%9Di%20gian%20x%E1%BA%A5p%20x%E1%BB%89%20b%E1%BA%B1ng%20nhau.>

[2] Phương Pháp Tính:

https://dutudn-my.sharepoint.com/:b:/g/personal/dtthoa_dut_udn_vn/EdwbYtB-OK1HpFVVges_4jABJjP8CoQfsvRi19P1Q-n8yA?e=mZl4RC

PHỤ LỤC

Code:

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#define MAX_SIZE 100

double determinant(double A[MAX_SIZE][MAX_SIZE], int n, int level) {
    double det = 0;
    double submatrix[MAX_SIZE][MAX_SIZE];

    if (n == 1) {
        return A[0][0];
    } else if (n == 2) {
        return A[0][0] * A[1][1] - A[0][1] * A[1][0];
    } else {
        for (int k = 0; k < n; k++) {
            for (int i = 1; i < n; i++) {
                int j;
                for (j = 0; j < n; j++) {
                    if (j < k) {
                        submatrix[i - 1][j] = A[i][j];
                    } else if (j > k) {
                        submatrix[i - 1][j - 1] = A[i][j];
                    }
                }
            }

            double sign = (k % 2 == 0) ? 1 : -1;
            double sub_det = determinant(submatrix, n - 1, level + 1);
            det += A[0][k] * sign * sub_det;
        }
    }
}
```

```
        // In ra quá trình tính định thức
        if (level == 0) {
            printf("Buoc %d: %.2lf * (%.2f) * det(submatrix)\n", k+1, A[0][k],
sign);
            for (int i = 1; i < n; i++) {
                for (int j = 0; j < n - 1; j++) {
                    printf("%6.2f ", submatrix[i - 1][j]);
                }
                printf("\n");
            }
            printf("\n");
        }
    }
}
return det;
}
```

```
void printMatrix(double A[][MAX_SIZE],int n) {
    int i,j;

    printf("Ma tran can tinh:\n");

    for ( i = 0; i < n; i++) {
        for ( j = 0; j < n; j++) {
            printf("%6.2lf ", A[i][j]);
        }
        printf("\n");
    }
}
```

```
int main() {
    int n,i,j;
    double A[MAX_SIZE][MAX_SIZE];
```

```
char input_file_name[100];
char output_file_name[100];
printf("TINH DINH THUC MA TRAN BANG PHUONG PHAP
LAPLACE\n\n");
// Nhập tên tệp tin đầu vào và kiểm tra tính hợp lệ
printf("Nhap ten file dau vao: ");
fgets(input_file_name, sizeof(input_file_name), stdin);
input_file_name[strcspn(input_file_name, "\n")] = '\0'; // Xóa ký tự newline

FILE* input_file = fopen(input_file_name, "r");
while (input_file == NULL) {
    printf("Loi:File khong ton tai. Nhap lai ten file: ");
    fgets(input_file_name, sizeof(input_file_name), stdin);
    input_file_name[strcspn(input_file_name, "\n")] = '\0'; // Xóa ký tự newline
    input_file = fopen(input_file_name, "r");
}

// Đọc kích thước ma trận và các phần tử từ tệp tin đầu vào
fscanf(input_file, "%zu", &n);
if (n <= 0 || n > MAX_SIZE) {
    printf("Loi:Ma tran sai dinh dang.\n");
    fclose(input_file);
    return 1;
}

for ( i = 0; i < n; i++) {
    for ( j = 0; j < n; j++) {

        fscanf(input_file, "%lf", &A[i][j]);
    }
}
fclose(input_file);
printMatrix(A, n);
```

```
printf("\n");
// Tính định thức của ma trận
double det = determinant(A,n,0);

// Nhập tên tệp tin đầu ra và ghi kết quả
printf("Nhap ten file dau ra: ");
fgets(output_file_name, sizeof(output_file_name), stdin);
output_file_name[strcspn(output_file_name, "\n")] = '\0'; // Xóa ký tự newline

FILE* output_file = fopen(output_file_name, "w");
if (output_file == NULL) {
    printf("Error:File khong ton tai.\n");
    return 1;
}

fprintf(output_file, "%.4lf\n", det);
fclose(output_file);

// In kết quả ra màn hình
printf("Dinh thuc ma tran can tinh = %.4lf\n", det);
printf("Ket qua duoc luu vao file %s\n", output_file_name);

return 0;
}
```