



**UNIVERSIDADE FEDERAL
DE MINAS GERAIS**

Departamento de Ciência da Computação

Algoritmos e Estruturas de Dados III- 2º Semestre 2014

Documentação do Trabalho Prático 0

Matrizes Complexas

Aluno : João Lucas Faria de Gonçalves
Número de Matricula : 2013048925

Professor: Marcos Augusto Menezes Vieira
Belo Horizonte
2014

Sumário

1.Introdução.....	1
2.Solução Proposta	1
2.1.Definição dos Módulo.....	3
3.Análise de Complexidade.....	3

1.Introdução

O trabalho consiste em implementar um programa para multiplicação de matrizes de números complexos. Número complexo é um numero $z = a + bi$ em que i representa a parte imaginaria , e os números a e b números reais do tipo double . O programa tem como objetivo familiarizar o aluno com os conceitos make , linguagem C e ambiente de programação Unix .

2 . Solução Proposta

A solução proposta foi a implementação de um tipo abstrato de dados para representar os números complexos, funções para manipular as matrizes e arquivos de entrada e saída de dados.

A implementação do programa foi estruturada em quatro módulos: arquivo.c , principal.c , multiplica.c e complexos.h .O módulo arquivo.c contém as funções de leitura e escrita de arquivos . O módulo principal contém o programa principal (main), as linhas de comando para o funcionamento e inicialização .O modulo multiplica.c contem as funções para operações básicas para o tipo estruturado números complexos e multiplicação de matrizes . Para gerenciar a compilação desses módulos foi usado um aplicativo make , nomeado de makefile . O funcionamento do programa foi todo estruturado no arquivo principal.c , onde ocorre as chamadas das funções . A logica do programa consiste em : ler os dados de um arquivo , alocar memoria dinamicamente para as matrizes que serão multiplicadas e matriz resultado e imprimir o resultado em um arquivo de saída .

2.1.Definições dos Módulos

- No módulo arquivo.c foram definidas as funções :

- void insere_complex_matriz(int linhas , int colunas , FILE *Dados , num_complex ** matriz)
- void imprime_output(int linhas , int colunas , num_complex **matriz , FILE *arquivo)

A função insere_complex_matriz() faz leitura dos dados no arquivo de entrada e insere o

valores lidos em uma matriz já alocada dinamicamente por um procedimento que será apresentado mais a frente .Ela recebe como parâmetro as dimensões da matriz , a matriz já alocada para receber os valores e um ponteiro do tipo FILE que aponta para o arquivo de entrada .

A função `imprime_output()` é responsável por fazer a saída do programa .Ela recebe as dimensões da matriz , a matriz resultado já com o resultado da multiplicação e um ponteiro do tipo FILE que aponta para o arquivo de saída .

- No módulo `multiplica.c` foram definidas as seguintes funções :

- `num_complex Insere_valor(double x,double y)`
- `num_complex Mult(num_complex a ,num_complex b)`
- `num_complex Soma(num_complex a , num_complex b)`
- `num_complex Sub(num_complex a , num_complex b)`
- `num_complex **Aloca_matriz_complex(int linha , int coluna)`
- `num_complex **Liberar_matriz(int linha, int coluna, num_complex **matriz)`
- `void imprime_matriz(int linhas , int colunas,num_complex **matriz)`
- `void mult_matriz(int linhas_matriz1,int colunas_matriz2, num_complex **mat1, num_complex **mat2, num_complex **result)`

As funções `Insere_valor()` e `Sub()` não foram usadas no programa , porém foram definidas por serem funções algébricas básicas para estrutura de dados número complexo , que podem ser aproveitadas na implementação de outros programas.

A função `Soma()` recebe dois números complexos (a e b) e retorna numero complexo (c) com o resultado da soma ,

A função `Mult()` recebe dois números complexos (a e b) e efetua a multiplicação dos números complexos respeitando a regra de multiplicação $(c + di)$ por $(e + fi) = (ce - df) + (de + cd)i$, e retorna o resultado dessa multiplicação .

A função `mult_matriz()` é a principal função do programa .Responsável pela multiplicação das matrizes essa função recebe como parâmetro valores da linha e coluna da matriz resultado , a matriz `mat1` e `mat2` para serem multiplicadas e a matriz `result` que armazena o resultado da multiplicação.

-No módulo principal.c foram definidas as rotinas de execução das funções para o funcionamento do programa de acordo com a estrutura do arquivo de entrada já pre estabelecida no problema .Ela obedece as sequências e a formas em que os dados serão lidos no arquivo de entrada , considerando primeiro o número de instâncias para saber quantas vezes será repetida a rotina de execução , em seguida as dimensões das matriz q será para alocar a memoria suficiente para armazenar os números complexos , após armazenar os valores nas duas matrizes será executado a multiplicação e impressão do resultado no arquivo de saída . Ao fim dessa rotina o programa libera o espaço de memoria alocado para as matrizes utilizadas no programa e se preciso executa todo processo novamente de acordo com o número de instâncias .

3-Análise de Complexidade

Algoritmos para manipulação de arquivos :

$\text{insere_complex_matriz}() = O(n^2)$

$\text{imprime_output}() = O(n^3)$

Algoritmos para manipulação e multiplicação de matrizes :

$\text{Aloca_matriz_complex}() = O(n^2)$

$\text{Liberar_matriz}() = O(n)$

$\text{insere_complex_matriz}() = O(n^2)$

$\text{zera_matriz_complex}() = O(n^2)$

$\text{mult_matriz}() = O(n^3)$

$\text{imprime_output} = O(n^2)$

A função $\text{mult_matriz}()$ apresenta maior custo no pior caso , sendo assim a sua ordem de complexidade predomina , porém devemos considerar o laço de repetição das instâncias no modulo principal.c .A repetição desse laço depende da entrada , logo a complexidade do programa será $O(n^4)$.