



DISCIPLINA: Algoritmos E Estrutura De Dados III

PROFESSOR: Thiago França Naves

Trabalho Prático – ESTRUTURA DE DADOS

Trabalho 1

Nome: Gabriel Carboneri Ra: 2576600

Nome: Igor Wingert Ra:2554080

§1.º - Introdução;

§2.º até §5.º - Explicação do código;

§6.º - Conclusão e desafios;

Neste trabalho, criamos uma matriz esparsa, na qual o usuário pode inserir um elemento em qualquer posição da matriz. Fizemos todas as funções solicitadas pelo professor e adicionamos uma função extra para mostrar a matriz transposta. Foram criadas 12 funções no total para a funcionalidade do código, além de duas estruturas para a montagem da matriz.

A estrutura da matriz, em relação aos ponteiros cabeça, foi pensada como uma lista circular, onde todos, no final, apontam para uma "cabeça mãe", que tem acesso às linhas e colunas. Cada ponteiro cabeça tem suas coordenadas e aponta para um elemento da matriz. Na parte dos elementos da matriz, cada nó tem um apontamento para cima, baixo, esquerda e direita, conseguindo apontar para outros iguais a ele em todas as direções. Cada nó também contém os dados a serem inseridos e suas respectivas coordenadas.

Na parte das funções, a primeira função, `criar()`, é responsável pela criação da "cabeça mãe", que é alocada dinamicamente e retorna pronta para ser usada. Já a segunda e a terceira, `gerar_colunas()` e `gerar_linhas()`, são parecidas, com a função de gerar as cabeças de linhas e colunas. Nessas funções, são criadas as listas circulares em que todas terminam na cabeça mãe. A quarta função, `tamanho_matriz()`, é responsável por gerar o tamanho desejado da matriz pelo usuário e utiliza as funções `gerar_colunas()` e `gerar_linhas()`.

A quinta função, e a mais importante para o funcionamento do código, é a de inserção dos elementos na matriz. O elemento é alocado e cria-se ponteiros para percorrer a lista de cabeças até encontrar as coordenadas correspondentes, fornecidas pelo usuário, para o elemento ser inserido. Após encontrar a linha e a coluna, o elemento é inserido na linha. Criam-se ponteiros auxiliares para percorrer a matriz, e, caso não haja nenhum elemento, a cabeça da linha aponta para o nó. Caso contrário, o nó aponta para os auxiliares que percorreram a linha e estão nas posições corretas para o encadeamento. Se o usuário tentar colocar um elemento em uma coordenada já existente, a função atualizará o valor daquela posição e liberará o novo nó. O processo na coluna é semelhante ao da linha.

A sexta função é a de impressão. Um auxiliar percorre toda a lista de linhas, e outro auxiliar imprime os valores de cada linha. A sétima função encontra um valor na matriz. Ela é parecida com a de impressão, mas, em vez de imprimir todos os nós, imprime apenas o nó com o valor correspondente ao solicitado pelo usuário e conta quantos números iguais existem. A oitava função libera a matriz inteira, tanto os elementos quanto as cabeças. A lógica para liberar os elementos é parecida com a de impressão, mas, neste caso, é criado mais um ponteiro auxiliar que aponta para o próximo elemento da matriz, liberando o nó anterior. Após liberar todos os elementos de uma linha, a cabeça da linha também é liberada. Quando todas as linhas e elementos são liberados, a lista de colunas é liberada da mesma forma, e, por fim, a cabeça mãe é liberada.

A nona função consulta um elemento pela sua coordenada, percorrendo as listas cabeça até encontrar a coordenada desejada. Um ponteiro auxiliar percorre os elementos até achar a posição correta. A décima função mostra os vizinhos de um elemento, seguindo a mesma lógica da consulta, mas, ao encontrar o elemento, o ponteiro aponta para os seus vizinhos e os imprime. A décima primeira função é o menu para o usuário, e a décima segunda função permite ao usuário informar a operação desejada. A última função é a da transposta, que segue a mesma lógica da função de impressão, mas imprime as colunas primeiro, formando assim a matriz transposta.

Os desafios e dificuldades do trabalho foram principalmente em relação às inserções, que levaram horas para alcançar um resultado final. A dificuldade de inserir os elementos em ordem aleatória sem causar bugs ou deixar de inserir algum elemento foi um desafio. Tivemos que modificar a estrutura da matriz para facilitar essa inserção, e o fato de os elementos e as cabeças terem coordenadas foi de grande ajuda. O desenho da matriz fornecido pelo professor no trabalho também ajudou muito a entender como os elementos deveriam ser dispostos.

Além da inserção, foi difícil começar, pois imaginar como a estrutura ficaria foi um pouco complicado. No início, fizemos uma matriz que, logo após ser criada, percebemos que não era uma matriz esparsa, e tivemos que recomeçar do zero. Em relação às outras funções, após a parte de inserção e estrutura estar consolidada, foi mais fácil implementar as funcionalidades de percorrer, imprimir e consultar a matriz, pois, antes disso, estava muito difícil fazer qualquer coisa funcionar corretamente.