

Introdução

O problema abordado por esse trabalho prático é que se deve implementar um grafo genérico, de forma em que ele deve fazer algumas funções a partir do usuário, assim as estruturas deverão ser montadas.

As funções do algoritmo serão as seguintes:

- Crie um grafo direcionado/não-direcionado com N vértices
- Adicione uma aresta: (u,v)
- Adicione N arestas: $(u_1,v_1),(u_2,v_2),\dots,(u_N,v_N)$
- Remova u
- Remova (u,v)
- Imprima a matriz de adjacência
- Imprima as listas de adjacência
- Existe o vértice u?
- Existe a aresta (u,v) ?
- Existe um caminho de u a v?
- Mostre um possível caminho de u a v caso exista.

Essas realizaram funções específicas de acordo com o desejo do usuário no arquivo.

Solução do Problema

Para a solução deste problema foram utilizados alguns tipos de dados abstratos, as funções utilizadas estarão descritas abaixo:

- Para a criação do grafo, independentemente se direcionado ou não, é reservado um espaço na memória, por meio do uso de struct chamada grafo, que contem um inteiro chamado vértice e um ponteiro para o grafo, chamada adj, que significa adjacente, e o calloc, para poder fazer a lista de adjacência, assim criando um vetor de ponteiro para cada vértice, e também alocando o espaço na memória para a matriz, usando ponteiros.
- É necessário que o grafo tenha um vértice inicial, e esse deveria ter seu endereço guardado para que o uso em outras funções não seja afetado. Ela cria um vértice e faz com que seu adjacente receba nulo, assim a primeira

etapa está concluída, para a representação de lista de adjacência, já para a matriz de adjacência, para cada aresta existente se coloca 1, caso não exista, 0.

- Cria vértice só irá ocorrer após a inicialização do grafo, ele já deveria existir, assim um novo vértice será acrescentado a cada chamada da função.
- Como o próprio nome diz o grafo direcionado tem uma observação, ele deveria seguir apenas um caminho a partir de cada vértice, quer dizer que ele mesmo estando conectado a outros dois ou mais, ele irá poder seguir apenas um caminho único. Assim a função cria grafo direcionado, cria o vértice e as arestas de acordo com a especificação do usuário.
- Verificar a aresta é uma função que funciona de forma que realiza um teste, esse compara a aresta pesquisada, assim até encontrar a aresta definida, caso verifique na matriz, apenas joga o número da posição (linha,coluna), caso seja a lista, ele percorre a lista específica do vértice até achar NULL.
- O algoritmo tem a opção de você adicionar uma ou N arestas, assim o usuário pode inserir uma nova aresta a partir do grafo já existente, modificando e alterando-o.
- Como existe a opção de adicionar, existe também a opção para remover a aresta ou o vértice se necessário, atribuindo o valor -1 para as arestas ou vértices removidos.
- A lista e a matriz serão limpas após o término da execução do programa.
- Existe uma função que serve apenas para exibir a lista ao usuário, essa percorre o grafo a partir da cabeça de cada lista (primeira posição de cada vetor), para que assim, cada vértice tenha definido seus adjacentes, exibindo os mesmos. E uma função para exibir a matriz, ele irá percorrer cada linha e coluna, exibindo assim a matriz.
- O grafo também tem o auxílio das matrizes, mas para isso o algoritmo cria uma matriz para que após isso a matriz seja preenchida de acordo com o grafo, e suas escolhas.
- A principal função de um grafo é definir o caminho, de forma que esse pode ser o melhor ou não, o mais curto ou não, assim o grafo pode definir maneiras de se chegar ao destino e novas possibilidades, por meio de

comparações e modificação de ponteiros, o grafo calculado e exibido ao usuário o caminho a partir do que foi desejado pelo usuário. A verificação do caminho descrito acima é feita por busca em profundidade, intuitivamente, o algoritmo começa num nó raiz (selecione algum nó como sendo o raiz, no caso de um grafo) e explora tanto quanto possível cada um dos seus ramos, antes de retroceder, onde a cada chamada da função do cálculo do caminho, todos os vetores auxiliares serão resetados.

- A função main abre o arquivo para sua leitura, nesse arquivo estarão as informações do usuário para os testes, ela lê cada parte do arquivo separada, assim cada função desejada entrará em um laço, e será executada. O usuário irá definir completamente como será montado o seu grafo.