Algoritmo de Busca em Largura (BFS)

Caroline de Oliveira Braga

[1]Centro Federal de Educação Tecnológica Celso Suckow da Fonseca (CEFET/RJ)
Unidade Descentralizada de Petrópolis – Petrópolis – RJ – Brasil

braga.coliveira@gmail.com

Resumo. *Este relatório visa explicar e mostrar o algoritmo de busca em largura (BFS) desenvolvido em linguagem C para a disciplina de algoritmos e estruturas de dados II.*

1. Introdução

Um algoritmo de busca percorre e examina todas as arestas de um grafo que são acessíveis a partir de um vértice em questão, chamado de origem. A diferença entre os algoritmos de busca consiste na ordem em que os vértices são visitados. Na busca em largura, depois de explorar a raiz, seus vizinhos são explorados e cada um desses vizinhos, tem os seus respectivos vizinhos explorados, até que todos os vértices alcançáveis a partir dessa origem sejam descobertos.

2. Desenvolvimento

O programa consiste na implementação de uma lista de adjacência, uma fila e o algoritmo de busca em largura.

2.1. Estruturas utilizadas

Para aplicar o algoritmo de busca em largura, o grafo foi representado como uma lista de adjacência, que foi implementada com uma lista encadeada. A representação do grafo foi feita com duas estruturas: Grafo e No. A estrutura Grafo contém o número de vértices do grafo e sua lista de vértices. Por sua vez, a estrutura No armazena o vértice em questão e aponta para o próximo vértice.

Além disso, foi implementada uma fila, utilizando a estrutura Fila, que armazena o seu início, topo, quantidade de elementos presentes na fila e uma lista dos elementos presentes na fila.

2.2. Funções

O programa utiliza a função main() para ler as informações do grafo, depois disso, a função inicializar() é chamada para inicializar o grafo, em seguida, a função add_aresta() é chamada para adicionar as arestas lidas da entrada ao grafo.

Com o grafo representado, é chamada a função bfs(), que executa a busca em largura. Nessa função, ocorre a criação de uma fila para armazenar os vértices a serem percorridos e ocorrem chamados para as seguintes funções: enfileirar(), que adiciona um vértice a fila, desenfileirar(), que remove um vértice da fila e vazia(), que verifica se a fila está vazia. Além disso, em cada execução da busca, é chamada a função saida(), que printa a saída formatada na tela do usuário, incluindo um chamado a função printar_fila(), que mostra a fila na tela.

2.3. O algoritmo

O algoritmo de busca em largura, implementado no programa utilizando a função bfs(), começa com a criação da fila que guardará os vértices a serem percorridos. Em segundo lugar, todos os vértices são definidos como da cor branca (W), os predecessores de cada vértice são definidos como -1 e a distância de todos para a origem é definida como ”inf”.

Em seguida, o algoritmo percorre a origem, atualizando sua distância para zero, sua cor para cinza (G), pois agora a origem já foi descoberta e enfileirando esse vértice. Após isso, o primeiro vértice da lista é desenfileirado e sua lista de vértices vizinhos, percorrida, até a lista ficar vazia. Ocorre uma verificação de cor para cada vértice da vizinhança e, se esse vértice for branco, sua cor é atualizada para cinza, sua distância para origem torna-se a distância do seu adjacente para origem mais um, e seu predecessor é o vértice que possibilitou a sua descoberta. Em seguida, esse vértice é enfileirado e o algoritmo segue para o próximo vizinho.

Quando um vértice tiver todos os seus vizinhos visitados, sua cor será alterada para preto (B) e, ao final dessa iteração, os dados referentes a essa execução do algoritmo serão mostrados na tela, por meio de um chamado a função saida().

3. Conclusão

O algoritmo é eficiente pois percorre todos os vértices do grafo a partir da origem e obtém os resultados desejados.