

Instituto Federal de Educação Ciência e Tecnologia de Brasília.

Alunos: Carlos Eduardo Pereira Santana; Danyelle da Silva Oliveira Angelo; Raquel Pinheiro da Costa;

Disciplina: Teoria dos Grafos - Professor: Raimundo Cláudio Vasconcelos

Trabalho Avaliativo 1

1 Instruções de compilação

Abra o terminal na pasta libGraph execute os seguintes comandos:

gcc -c nomeArquivo.c

Faça isso para todos os arquivos .c, esse comando irá gerar arquivos objetos (.o), depois que estes arquivos tiverem sido gerados, digite o comando abaixo:

ar rcs nomeBiblioteca.a arquivo1.o arquivo2.o ...

2 Instruções para execução

Na pasta raiz do programa (a biblioteca já deve ter sido gerada) execute os comandos(em negrito) abaixo:

gcc main.c -o nomeSaida nomeBiblioteca.a

Indique o caminho da biblioteca, caso ela esteja em outra pasta.

./nomeSaida

3 Arquivos

- Diretório raiz
 - main.c : arquivo principal;
 - busca_listAdj.txt : arquivo usado para montar árvores de busca para lista de adjascência;
 - busca_MatAdj.txt :arquivo usado para montar árvores de busca para matriz de adjascência;
 - grafo.txt : arquivo de entrada, contém as informações do grafo (número de vértices, e os pares de vértices);

- saida.txt : arquivo de saída com as informações do grafo (número de arestas e vértices, grau de cada vértice).

- libGraph

Para cada um dos arquivos .c listados abaixo temos um arquivo objeto (.o)

- graph.c : contém a função de inicialização do grafo, a que preenche o arquivo saida.txt, as funções de busca (para todas as representações) e o algoritmo de dijkstra;
- list.c : contém as funções de lista e pilha;
- listAdj.c : monta a lista de adjacência propriamente dito;
- matAdj.c: monta a matriz de adjacência;
- graph.h, list.h, listAdj.h, matAdj.h: contrato dos arquivos .c que recebem o seu nome, este arquivo apresenta os protótipos das funções e um apanhado geral de cada função implementada, tudo que é declarado nele, deve estar no arquivo .c associado a ele.
- libGraph.a : biblioteca com as funções implementadas nos arquivos citados acima, na hora de compilar um arquivo que precisa usar as funções dos arquivos citados acima, basta compilar o arquivo em questão junto a essa biblioteca.

4 Estudo de Caso 1

Considere o grafo de colaboração entre pesquisadores disponível no site da disciplina. Utilizando este grafo, responda às perguntas abaixo.

1. Compare o desempenho em termos de quantidade de memória utilizada das duas representações do grafo. Determine a quantidade de memória (em MB) utilizada pelo seu programa quando você representa o grafo utilizando uma matriz de adjacência e lista de adjacência.

Resposta: Após compilar o arquivo principal junto a nossa biblioteca, rodamos o arquivo compilado junto a aplicação valgrind (que detecta erros decorrentes do uso incorreto da memória dinâmica). De acordo com o programa para a representação na forma de lista de adjacência (opção 2), o programa construído utiliza 7.384.556 (8 MB) bytes da memória, já no caso da matriz de adjacência (opção 1), o programa retorna (para a entrada em questão) o texto "morto", não sabemos ainda o porquê desse retorno.

2. Compare o desempenho em termos de tempo de execução das duas representações do grafo. Determine o tempo necessário para executar uma busca em largura em cada um dos casos (considere o pior caso).

Resposta: *A lista de adjascência possui desempenho de tempo melhor do que a de matriz de adjascência para os dados do arquivo Collaboration graph (Usamos como vértice inicial o vértice 2).*

- *Matriz de adjascência: Tempo gasto = 10067.1 ms*
- *Lista de adjascência: Tempo gasto = 79.437 ms.*

3. Obtenha os componentes conexos do grafo. Quantos componentes conexos tem o grafo? Qual é o tamanho do maior e do menor componente conexo?

Resposta:

5 Estudo de Caso 2

Considere o grafo de conexão das redes que formam a Internet (AS Graph). Utilizando este grafo, responda às perguntas abaixo.

1. Trace um gráfico com seu resultado. Qual é o maior grau do grafo? E o menor? Como isto se compara ao maior grau possível?

Resposta: *O maior grau é 2161, e o menor é 1. Assim é possível concluir que o grafo não é conexo.*

2. Obtenha os componentes conexos do grafo. Quantos componentes conexos tem o grafo? Qual é o tamanho do maior e do menor componente conexo?

Resposta:

3. Faça uma busca em largura a partir do vértice 1. Neste caso, o maior nível da árvore geradora de busca representa a maior distância do vértice 1 a qualquer outro. Determine este valor. Repita este procedimento para outros vértices. O que você pode concluir?

Resposta: *Conclui-se que o maior diâmetro é 7.*

4. Determine o diâmetro da Internet. Lembrando que o diâmetro é a maior distância entre qualquer par de vértices do grafo (ou seja, o comprimento do maior caminho mínimo do grafo). Utilize a BFS para responder esta pergunta.

Resposta: *O diâmetro é 7 .*