

SCC0216 - Modelagem computacional em grafos
Professor: Alneu de Andrade Lopes
Estagiário PAE: Alan Valejo

LAB 2: Busca

1. Especificação

O laboratório consiste na implementação de busca em dígrafos não-ponderados. Deverá ser utilizada a linguagem de programação C. As buscas consideradas são as buscas em largura e em profundidade.

2. Descrição da entrada

O programa principal deverá ler da entrada padrão a lista de arestas do dígrafo e representá-lo em memória - matriz ou lista de adjacência, conforme desejar. Em seguida, o programa deverá ler instruções, encontrar um caminho entre dois vértices através de determinada tipo de busca e imprimir o resultado na saída padrão.

Na primeira linha da entrada, haverá o descritor do dígrafo contendo 2 números separados por espaço. Os números indicam, nesta ordem, o número de vértices e de arestas do dígrafo. Nas linhas seguintes, as arestas do dígrafo serão representadas por dois números indicando os vértices de origem e de destino.

Por fim, as buscas serão representadas por duplas contendo o vértice de origem e o vértice de destino. Para cada instrução, o programa deverá realizar a busca e imprimir o caminho encontrado (se existir) entre os vértices de origem e de destino. O caminho deverá ser impresso contendo o índice do vértice de origem, os índices dos vértices intermediários e, por fim, o índice do vértice de destino. Caso não exista caminho entre os vértices, deixe a linha em branco.

3. Dicas

- Utilizem o vetor antecessor (consulte o material de aula) para encontrar o caminho até um vértice.
- O sistema considerará que a expansão dos nós são feitas por ordem do índice, portanto, se utilizarem lista de adjacência, mantenha a lista de vértices adjacentes ordenada.
- Utilizem o TAD fila (do semestre passado) para implementar a busca em largura.

4. Submissão

O exercício deverá ser entregue pelo sistema run.codes . Todos os alunos deverão submeter seus códigos no 'Exercício 2: Busca em Largura' até o final da aula. Somente a última submissão será considerada. Todas as demais submissões serão desconsideradas, incluindo aquelas dentro do período

normal de submissão. Os exercícios deverão submetidos em um arquivo .zip contendo código-fonte do programa e um Makefile para compilação e teste do trabalho (verificar com o estagiário PAE, caso não saiba escrever um Makefile). Se necessário, incluam no .zip um arquivo chamado readme com informações que julgarem necessárias.

Os códigos deverão ser compilados pelo compilador gcc com a flag -std=c99. A não conformidade das implementações com a versão C estabelecida acarretará em nota zero. Atenção! Todos os códigos enviados passarão pelo sistema de verificação de plágio. Se forem identificados códigos duplicados, todos os alunos envolvidos receberão nota zero.

5. Correção e Avaliação

As implementações serão avaliadas por meio de casos de testes, com peso 7, e pela legibilidade e boas práticas de programação, com peso 3.

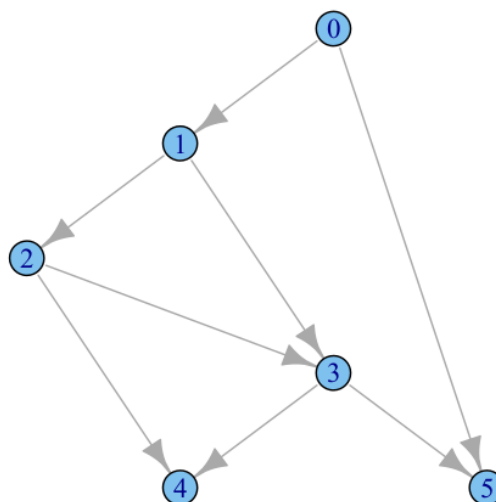
Os seguintes casos implicarão em nota zero:

- Não conformidade com a versão C99;
- Programas não estruturados como um TAD;
- Exercícios plagiados.

Apêndice: Exemplos

Os comentários são apenas descritivos. Estes não existirão nas entradas e nem deverão ser impressos como saída.

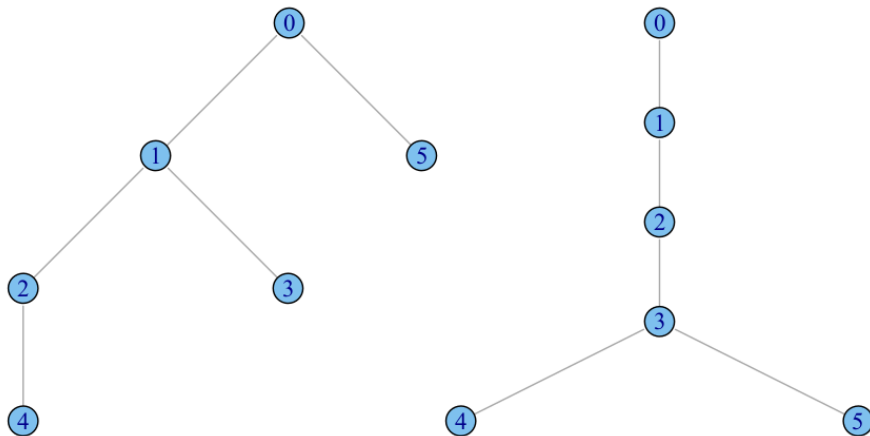
Dado o digrafo:



Entrada

```
6 8          // 6 vértices e 8 arestas
0 1
0 5
1 2
1 3
2 3
2 4
3 4
3 5
4 0
0 0
0 4
0 5
```

Partindo do vértice 0, as árvores geradas, respectivamente, pelas buscas em largura e em profundidade são:



Deste modo, utilizando a busca em largura, o programa terá como saída

```
// de 4 para 0, não há caminhos
0          // de 0 para 0, já está no vértice de destino
0 1 2 4    // caminho de 0 para 4 encontrado na busca em largura
0 5        // caminho de 0 para 5 encontrado na busca em largura
```

Utilizando a busca em profundidade, a saída

```
// de 4 para 0, não há caminhos
0          // de 0 para 0, já está no vértice de destino
0 1 2 3 4   // caminho de 0 para 4 encontrado na busca em profundidade
0 1 2 3 5   // caminho de 0 para 5 encontrado na busca em profundidade
```