

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

LAB 3: Ordenação Topológica

1. Especificação

O laboratório consiste na implementação de um algoritmo de ordenação topológica em dígrafos não-ponderados acíclicos. Deverá ser utilizada a linguagem de programação C99. A ordenação topológica deverá ser realizada pela travessia 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 executar a travessia em profundidade, encontrar a ordenação topológica 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.

A ordenação deverá ser impressa com os índices dos vértices separados por espaço. Após imprimir a ordenação, insira uma quebra de linha.

3. Dicas

- O vetor *f* do algoritmo de busca em profundidade pode ser utilizado para encontrar a ordenação topológica.
- 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.

4. Submissão

O exercício deverá ser entregue pelo sistema run.codes. Todos os alunos deverão submeter seus códigos no 'Exercício 2: Ordenação topológica' 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:

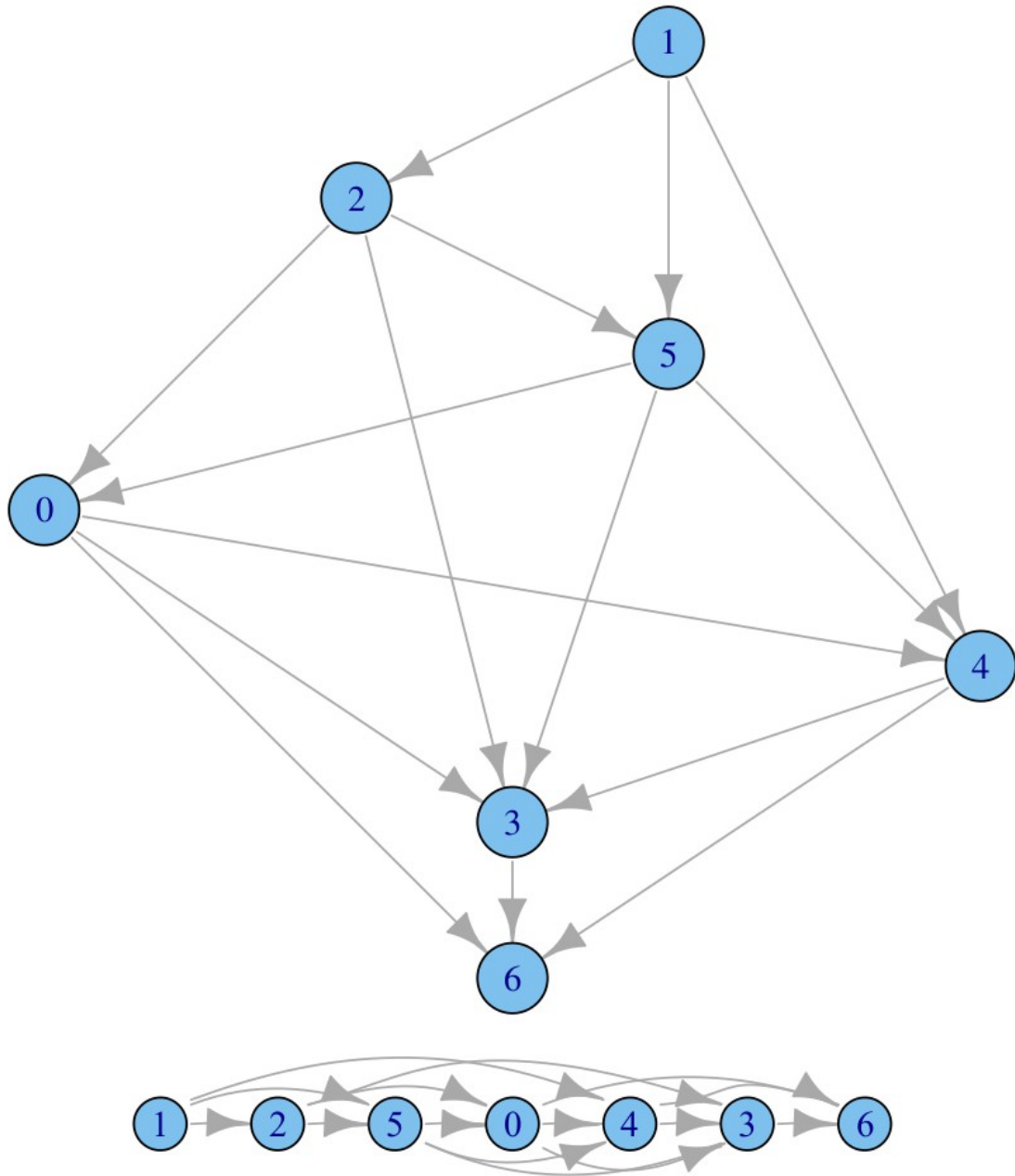
- Não conformidade com a versão C99;
- Programas não estruturados como um TAD;
- Exercícios plagiados.

Apêndice: Exemplos

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

Dado o digrafo:

7	15
2	5
5	0
2	0
0	3
4	6
0	4
1	5
4	3
1	2
5	4
0	6
1	4
5	3
3	6
2	3



Deste modo, o programa terá como saída

| 1 2 5 0 4 3 6