Universidade Estadual de Maringá
Centro de Tecnologia – Departamento de Informática
Algoritmos em Grafos
Prof. Daniel Kikuti

Primeiro Trabalho

1 Objetivos

Os objetivos deste trabalho são:

- Revisar conteúdo referente à primeira avaliação.
- Verificar o aprendizado dos alunos através da implementação de estruturas de dados para representação de grafos e de diversos algoritmos vistos em aula.
- Desenvolver habilidades de identificação e resolução de problemas em aplicações que envolvem grafos.

2 Informações gerais

- Este trabalho compõe a primeira avaliação periódica e possui peso 2,0.
- Formar equipes com no máximo três alunos.
- Os algoritmos poderão ser codificados em quaisquer linguagem de programação.
- O compartilhamento de informações entre equipes é permitido, mas compartilhar códigos não.
- Trabalhos que tenham porções significativas de código iguais, ou copiados da internet, serão anulados.
- Se usar algum material como referência, cite-o dentro do código fonte como comentário.

3 O que deverá ser feito?

1. Implementar os seguintes algoritmos:
   - Busca em largura (bfs). Entrada: grafo orientado e um vértice inicial s. Saída: para cada vértice, imprimir o vértice, seu predecessor na árvore de busca em largura e sua distância de s.
   - Busca em profundidade (dfs). Entrada: grafo orientado. Saída: para cada vértice, imprimir o vértice, tempo de descoberta, tempo de término e predecessor na floresta de busca em profundidade.
   - Ordenação topológica. Entrada: grafo orientado. Saída: imprimir uma linha contendo uma ordenação topológica com os vértices separados por um espaço.
   - Componentes fortemente conexos (scc). Entrada: grafo orientado. Saída: imprimir cada scc em uma linha com os vértices separados por espaços.
   - Bellman-Ford. Entrada: grafo orientado com pesos nas arestas e um vértice inicial s. Saída: para cada vértice, imprimir o vértice, seu predecessor e sua distância de s.
   - Dijkstra. Entrada: grafo orientado com pesos nas arestas e um vértice inicial s. Saída: para cada vértice, imprimir o vértice, seu predecessor e sua distância de s.
   - Floyd-Warshal. Entrada: grafo orientado com pesos nas arestas. Saída: imprimir a matriz de distâncias.
2. Adaptar os algoritmos para resolver um dos seguintes problemas:
   - No site da Uva Online Judge (https://uva.onlinejudge.org): 336, 439, 532, 11686 e 10305.
   - No site da URI (www.urionlinejudge.com.br): 1082, 1128, 1100, 1442, 1454, 1479 e 1081.

3.1 Entrada de dados

Para realizar os testes dos algoritmos implementados, assuma que o grafo será informado na entrada padrão, segundo o formato `.tgf` (trivial graph format).

A Figura 1 a seguir apresenta um exemplo. A entrada consiste de um conjunto de vértices e um conjunto de arestas separados por um caractere `'#'`. No conjunto de vértices, cada linha contém um par, indicando o vértice e seu rótulo. No conjunto de arestas, cada linha contém um par indicado que há uma relação entre os dois vértices. Neste formato de arquivo, a interpretação do tipo do grafo (orientado ou não) fica a critério da aplicação. Se as arestas contêm pesos, isto é representado adicionado-se um terceiro item na linha. Desta forma, a linha 1 2 10 indica que a aresta (1,2) tem peso 10.

```
1 a
2 b
3 c
4 d
#
1 2
2 3
2 4
```

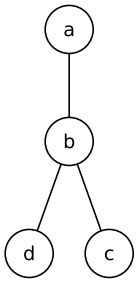

Figura 1 - Dados inseridos na entrada padrão segundo o formato .tgf e sua representação gráfica.

4 O que será avaliado no trabalho?

O trabalho será avaliado de acordo com os critérios:

- Correção da implementação.
- Clareza e organização do código.
- Tempo de execução de acordo como os algoritmos estudados em aula.
- Exemplos desenvolvidos e usados como teste.

5 Apresentação e entrega do trabalho

A data de apresentação e entrega do trabalho será definida em aula e divulgada no Moodle.