Universidade Federal de Lavras

Ciência da Computação

Prof. Douglas Aquino T. Mendes

Algoritmos em Grafos

Atividade de Implementação

1. **Objetivo:**

   Implementar algoritmos para análise de propriedades de grafos, avançando gradualmente à medida que a disciplina progride.

2. **Descrição:**
   a. Os alunos serão desafiados a implementar um **sistema modular** para análise de propriedades de grafos armazenados em arquivos de texto. **A implementação será progressiva**, com novas funcionalidades adicionadas conforme o avanço da disciplina. A estrutura básica consistirá em um menu interativo onde o usuário poderá selecionar a propriedade do grafo que desejam analisar.

   b. Os alunos são livres para decidir a linguagem de programação a ser utilizada.

   c. Os alunos possuem 60 dias para implementar as funções sugeridas, e são livres para adicionar qualquer outra a qual considere relevante/interessante.

   d. A avaliação se dará em cima da porcentagem de funções requeridas implementadas, por seção do menu, a qual possui 5 seções. Cada seção vale 20% da nota do trabalho.
      - Funções extras ultrapassam o valor total do trabalho
      - Exemplo:
        - Para ter 20% da nota na seção X, deve-se implementar 5 funções, caso faça 6, essa seção

passa a valer 24% (pois cada função nessa seção vale 4%). O trabalho passa a valer 104%.

- A nota total do trabalho não excede 113%.

e. A atividade poderá ser realizada em trios, duplas ou sozinho.

f. Todo código desenvolvido deverá ser enviado para um repositório de códigos online, **GitHub**, com o devido *commit* resumindo o que foi implementado na versão atual.
   - https://github.com/git-guides/git-commit
   - Os commits serão utilizados para averiguação de participação dos alunos. Alunos que não participarem na implementação do código, não receberão pontuação.
   - O usuário SrAquino (e-mail: douglasaquino4@gmail.com) deverá ser adicionado como colaborador do projeto.

g. Todas as funções devem possuir um comentário javadoc
   - /** Essa função recebe a entrada em tal formato e responde tal coisa */
     - https://pt.stackoverflow.com/questions/6733/para-que-serve-o-c%C3%B3digo-na-linguagem-java

h. O grafo a ser analizado estará em um arquivo .txt, e deve ser representado como no exemplo a baixo, seguindo as regras de chaves ‘{‘ ‘}’ para representar o conjunto, e finalizar com ponto e virgula ‘;’, as arestas só precisam ser representadas uma vez em caso de grafos bidirecionais:

V = {1,2,3,4,5}**;** A = {(1,2),(2,3),(3,1),(4,5)}**;**

Onde V são os vértices e A são as arestas do grafo. Essa padronização será importante para os testes de avaliação.

3. **Etapas:**
    a. Implementação básica:
        - O programa deve perguntar se o grafo é direcionado ou não.
            - Caso seja não direcionado, o programa deve adicionar as arestas.

        - O programa deve fazer a leitura de um arquivo .txt
            - Identificar os vértices e arestas.
            - Retornar erro caso algo esteja incorreto na representação do grafo no arquivo
                - Exemplos:
                    i. V = { 1, 2; A = {};
                    ii. V = {1,2,3,4}; A = {12, 34};
                    iii. V = {1,2,3}; A = {(1,8)}

        - O programa deve ser modular, e ao chamar cada função disponível no menu, o programa deve mostrar o tempo que levou para executar a função.
            - https://www.youtube.com/watch?v=9GaDSKZ3sM4
            - Para comparar a eficiência de algoritmos com resultados equivalentes, como BFS e DFS em grafos esparsos e densos.

    Menu:
        a. Representações
            i. Matriz de Adjacência
            ii. Lista de Adjacência
            iii. (Desafio) Representação gráfica
                1. **https://www.youtube.com/watch?v=PfT8_2sKReo**

        b. Remoções e Inserções
            i. Arestas
                1. Remover
                2. Adicionar
            ii. Vértices
                1. Remover
                2. Adicionar

(Ao finalizar essas operações, o arquivo txt deve ser atualizado)

c. Verificações
    i. Quantidade de vértices
    ii. Quantidade de arestas
    iii. Grau de um vértice
    iv. O grafo é conexo?
    v. O grafo é fortemente conexo? (Essa opção só deve aparecer caso o grafo seja direcionado)
    vi. O grafo possui ciclos?
    vii. O grafo é Euleriano? (A resposta deve ser Euleriano, Semi-Euleriano ou Não)

d. Árvores
    i. Busca em Largura
    ii. Busca em Profundidade
    iii. Geradora mínima
        1. Kruskall
        2. Prim

e. Algoritmos
    i. Ordenar topologicamente
        1. Kahn
        2. DFS para ordem topológica
    ii. Identificar componentes forte
        1. Kosaraju

[!] Novas funcionalidades serão solicitadas ao decorrer das aulas.

Cronograma (26/04 à 26/06):

- A progressão na atividade será verificada a cada 10 dias:
  - 06/05 (Inicialização do projeto e representações do grafo)
  - 16/05 (Remoções e inserções)
  - 26/05 (Verificações e Árvores)
  - 05/06 …
  - 15/06 …
  - 25/06 …
  - 24/06 à 29/06 (Apresentação dos trabalhos)

Horários de atendimento: (Agendar)

Quartas das 17:00 às 19:00

Quintas das 17:00 às 21:00

E-mail: douglas.mendes@ufla.br