

Trabalho Prático

Vencimento 24 nov por 23:59

Pontos 20

Enviando um upload de arquivo

Disponível até 24 nov em 23:59

O trabalho pode ser feito em grupos de até 5 alunos. As implementações podem ser feitas nas linguagens C/C++, C#, Java, e Python, e o relatório deve ser feito em LATEX. Ele deverá ser entregue no Canvas, até as 23:59 da data limite estipulada (data mostrada na tarefa do Canvas). Cópias serão sumariamente zeradas. Cada um dos membros do grupo deverá entregar uma cópia do trabalho (alunos que não entregarem, mesmo que outros membros do grupo entreguem, receberam nota igual a zero). A entrega deve ser realizada da seguinte forma: um arquivo .zip ou .rar contendo o código fonte do trabalho, o fonte do relatório (.tex) e o PDF do relatório. Além disso, indiquem no relatório as responsabilidades e o que foi feito por cada membro do grupo.

O trabalho será dividido em três partes, todas com a mesma data de entrega.

Parte 1

Desenvolver uma biblioteca para a manipulação de grafos, contendo:

Representações

- Representação de grafos utilizando Matriz de Adjacência
- Representação de grafos utilizando Matriz de Incidência
- Representação de grafos utilizando Lista de Adjacência

Manipulação

Funções básicas para manipulação de grafos em ambas representações, incluindo:

- Criação de um grafo com X vértices (o número de vértices deve ser inserido pelo usuário)
- Criação e remoção de arestas
- Ponderação e rotulação de vértices
- Ponderação e rotulação de arestas
- Checagem de adjacência entre vértices
- Checagem de adjacência entre arestas
- Checagem da existência de arestas
- Checagem da quantidade de vértices e arestas
- Checagem de grafo vazio e completo
- Checagem de conectividade em simplesmente conexo, semi-fortemente conexo e fortemente conexo
- Checagem de quantidade de componentes fortemente conexos com Kosaraju
- Checagem de ponte e articulação

Parte 2

Uma ponte em um grafo é definido como uma aresta cuja remoção desconecta o grafo. O problema de se determinar pontes existentes em um grafo apresenta várias aplicações, dentre elas encontrar caminhos (ou ciclos) eulerianos. Na segunda etapa deste trabalho você deverá implementar dois métodos para identificação de pontes: (i) método naive em que testa-se a conectividade de um grafo para cada remoção de aresta (utilizando uma busca em largura ou profundidade por exemplo); e (ii) método baseado em Tarjan (artigo em anexo). Após implementadas as duas soluções para detecção de pontes, você deverá encontrar um caminho euleriano, usando Algoritmo de Fleury, em um grafo euleriano usando as duas estratégias implementadas. Ilustre os tempos computacionais necessários para as duas estratégias utilizando como teste grafos aleatórios simples contendo 100, 1000, 10000 e 100000 vértices.

Parte 3

Após terminadas as entregas 1 e 2, implementar a funcionalidade de ler e salvar os grafos em arquivos, em um dos tipos suportados pelo software de visualização de grafos Gephi. Os formatos suportados são: GEXF, GDF, GML, GraphML, Pajek NET, GraphViz DOT, CSV, UCINET DL, Tulip TPL, Netdraw VNA, Spreadsheet. A documentação de cada formato pode ser encontrada em: <https://gephi.org/users/supported-graph-formats/>
Com a nova funcionalidade de exportação de grafos criados em sua ferramenta, gerar as ilustrações de exemplo presentes na documentação utilizando a ferramenta Gephi.

Relatório:

Entre 12 e 15 páginas;

Demonstrar os resultados obtidos e telas produzidas;

Em formato de artigo SBC latex:

<https://www.overleaf.com/latex/templates/sbc-conferences-template/blbxwjwzdngr>

Apresentação em vídeo (demonstração) e presencial (arguição):

Demonstrar os resultados;

Todos os membros do grupo devem participar;

Duração da apresentação entre 5 e 10 minutos.

Obs: Somente serão avaliadas entregas que tenham todos os itens, a falta de um deles causa nota igual à zero.