

Trabalho Prático 1

Professor Mayron Moreira
Universidade Federal de Lavras
Departamento de Ciência da Computação
GCC218 - Algoritmos em Grafos
Valor máximo: 15 pts

5 de julho de 2017

1 Descrição do problema

Pesquisas recentes mostram que até janeiro de 2016, foram aceleradas até 1100 *startups* no Brasil [1]. Este fenômeno é impulsionado, em grande parte, por um empreendedorismo que agrega valor a produtos com tecnologias capazes de servir de apoio às mais complexas tomadas de decisão. No ambiente acadêmico, é importante que incentivemos os estudantes em tais iniciativas, a fim de produzir inovação alicerçada a uma boa base técnica.

Impulsionados pelos problemas resolvidos na prática via redes complexas ou grafos [3], você e mais alguns colegas de turma estão interessados em abrir uma *startup* cujo principal produto consiste em um *software* que, dado um grafo qualquer, apresente um conjunto de características deste, a fim de auxiliar o seu cliente nas diversas tomadas de decisão. Em uma rede metroviária, por exemplo, o responsável pela sua administração poderia demandar ao sistema informações tais como quais são as estações que interligam todas as outras (direta ou indiretamente), a alcançabilidade dos usuários em distintos pontos da cidade, estações mais carregadas, entre outros dados.

Uma vez que a competitividade é inerente ao mercado de trabalho, você e seus sócios pactuam que o sistema produzido por vocês será o melhor dentre todas as empresas concorrentes. A criatividade é um fator necessário para o sucesso de sua *startup*, porém, não é o suficiente. Seu produto deve ter um bom embasamento técnico e científico. Após várias rodadas de *brainstormings*, os requisitos básicos do sistema foram elencados e deverão ser implementados, como mostrado a seguir.

2 Detalhes de implementação

2.1 Estrutura de dados

Seu programa deve ler um grafo e armazená-lo em uma estrutura de dados apropriada definida pelo usuário. Como opções, temos:

- Matriz de adjacências;
- Matriz de incidências;
- Lista de Adjacências

Caso o usuário queira mudar a representação do grafo, seu programa deverá realizar a conversão automática para a nova estrutura selecionada.

2.2 Algoritmos

Características a respeito dos vértices e da forma com que se relacionam podem ser fundamentais para detecção de pontos críticos em uma rede. Em um grafo não-direcionado, existem algumas perguntas que um sistema em tecnologia de informação poderia responder:

P1 Se for conexo, qual aresta que se retirada o torna desconexo? Existe apenas uma ou mais arestas com essa peculiaridade?

P2 Existe um único vértice que, se retirado, causaria uma desconexão no grafo?

De um modo geral, tanto para redes orientadas ou não-orientadas:

P3 Partindo de um vértice qualquer, quantos outros vértices podemos alcançar no grafo?

P4 Qual é o diâmetro (maior entre as menores distâncias) do grafo? (Observação: para cada par de vértices, os considere distintos.)

P5 O grafo é desconexo, (fracamente) conexo, semi-fortemente conexo ou fortemente conexo?

P6 Quantos vértices e quantas arestas teria o grafo simplificado de um grafo direcionado se agrupássemos cada componente fortemente conexa deste em um vértice?

3 Problemas teste

Os arquivos de teste estão na pasta *instances.zip*, disponível no *Campus Virtual*. Note que os arquivos das instâncias “metrô” (grafos não-direcionados) poderão conter arestas nos dois sentidos, isto é, (u, v) e (v, u) . Isso se deve a uma particularidade na leitura destes arquivos, não interferindo no número real de arestas do grafo.

4 Regras para a entrega

- Os alunos farão os trabalhos em grupos de no máximo 3 pessoas.
- Não serão permitidos trabalhos individuais.
- O trabalho deverá ser implementado em C++ ou Python.

- Cada grupo deverá implementar a estrutura ou classe Grafo.
- Os grafos serão orientados ou não-orientados.
- As perguntas às quais o usuário busca a resposta devem ser parametrizadas pelo programa. Assim, uma execução do tipo `python TP1.py nomeArquivo -p1 -p2 -p3 0 1 -p6` deve imprimir na tela: (i) quantas arestas que retiradas isoladamente deixam o grafo não-orientado (se for o caso) desconexo; (ii) 1 (se existe este vértice) ou 0, caso contrário; (iii) quantos vértices são alcançáveis a partir do vértice 0 e do vértice 1; (iv) a quantidade de vértices e de arestas do novo grafo com as componentes agrupadas.
- **Data e hora de entrega: 15/07/2017, até às 23h55.**
- **Qualquer constatação de cópia ou plágio de trabalhos acarretará em nulidade das notas de todos os membros dos grupos envolvidos.**
- O *upload* do código fonte referente a este trabalho deve ser feito no Campus Virtual, na respectiva sala da disciplina dos membros do grupo, em local devidamente especificado. Além disso, em um arquivo .pdf de no máximo 10 páginas, cada grupo deverá entregar um documento, dividido em 2 seções, que deverá constar:
 - **Algoritmos implementados:** apresentação dos algoritmos que respondem cada pergunta, via pseudocódigo, e sua devida explicação. O grupo deve explicar qual foi a estratégia e a lógica adotada no método proposto, devidamente justificadas.
 - **Resultados:** apresentação de uma tabela de resultados que respondam as perguntas P1, P2, P4, P5 e P6. Nesta última, o número de vértices e arestas do novo grafo devem ser apresentados. Os membros do grupo devem indicar o tempo computacional (em segundos) gasto para responder todas as perguntas, em cada instância. Por fim, escolha uma instância relativa à rede social e uma instância de uma rede de transporte e interprete os resultados obtidos, para cada um dos dois casos. O que significa encontrar a aresta que responda P1, por exemplo?
- Apenas um dos alunos do grupo deve depositar o conteúdo do trabalho do sistema. Para tanto, a pasta compactada com todos os arquivos deve ter o seguinte formato:

$$matriculaAluno1_matriculaAluno2_matriculaAluno3.rar$$
- Os grupos que fizerem o trabalho em C++ deverão fornecer um Makefile para compilação do código.
- Será descontado 1 ponto dos membros de cada grupo cujo trabalho não seguir as especificações estabelecidas neste documento.
- Soluções do tipo força bruta não serão consideradas.
- Trabalhos entregues após a data limite terão os seguintes descontos de nota:
 - 09/07: 20% da nota;

- 10/07: 40% da nota;
 - 11/07: 60% da nota;
 - 12/07: 80% da nota;
 - Após dia 12/07: não serão aceitos.
- Clareza e organização do código fonte serão itens levados em conta na avaliação.
 - O docente tem total liberdade de solicitar a qualquer grupo, pessoalmente, esclarecimentos sobre o trabalho implementado.

Referências

- [1] Abreu, P. R. M., Campos, N. M. (2016). O panorama das aceleradoras de startups no Brasil. *CreateSpace Independent Publishing Platform*, USA, 48p.
- [2] Derrible, S. (2012). Network centrality of metro systems. *PloS one*, 7(7), e40575.
- [3] Latora, V., Nicosia, V., Russo, G. (2017). Complex Networks: Principles, Methods and Applications. *Cambridge University Press*, UK, 594p.