UNIVERSIDADE DE SÃO PAULO
INSTITUTO DE CIÊNCIAS MATEMÁTICAS E DE COMPUTAÇÃO
DEPARTAMENTO DE CIÊNCIAS DE COMPUTAÇÃO

SCC 603 - Algoritmos e Estruturas de Dados II - 2º semestre 2017
Profa. Elaine Parros M. de Sousa

Trabalho 1 – Grafos

1. Objetivo do Trabalho:

Aplicar os conceitos estudados na disciplina de Algoritmos e Estrutura de Dados II para implementação de soluções para problemas práticos.

Neste trabalho, o aluno deverá desenvolver um programa em linguagem C ou C++ para resolução do problema especificado na **seção 4** utilizando conceitos de grafos e estruturas de dados.

O programa em C/C++ deve ser compilado no **Code::Blocks 16.01**.

2. Critérios de Avaliação

O programa a ser desenvolvido neste projeto deve seguir rigorosamente os formatos de entrada e saída definidos.

Implemente seu trabalho com seus colegas de grupo sem compartilhar, olhar código dos outros grupos, ou buscar na Internet.

- A nota atribuída ao trabalho será de zero a dez inclusive.
- A nota máxima é atribuída se o trabalho avaliado atender todos os requisitos e estiver bem organizado e apresentado.
- Para cada requisito não atendido no trabalho, será descontada uma pontuação da nota até o limite mínimo da nota.
- Será atribuída nota ZERO quando:
    - o trabalho não for submetido até a data máxima possível;
    - ou/e o trabalho apresentar muitos erros;
    - ou/e não compilar;
    - ou/e for detectado plágio do trabalho (parcial ou total);
    - ou/e, não for entregue relatório conforme especificação.

Os itens de avaliação dos programas são:

1. (40%) O programa funciona corretamente nas tarefas descritas para todos os casos de teste e processamento correto das entradas e saídas;
2. (30%) Qualidade da solução desenvolvida, estruturação do código e implementação dos Tipos Abstratos de Dados, bom uso das técnicas de programação e eficiência da solução em termos de espaço e tempo;
3. (10%) Boa indentação e uso de comentários no código, além de boa estruturação e modularização;

4. (20%) Documentação externa: **relatório** curto (em pdf) explicando o tipo de grafo usado na modelagem (direcionado, não direcionado, ponderado, não ponderado, conexo, não conexo, completo, simples, etc…), as estruturas de dados, algoritmos e técnicas utilizados para implementar a solução do problema. Explique a lógica para resolução do problema. Ilustre o grafo definido na modelagem.

O programa deve ser desenvolvido em grupos de 3 ou 4 alunos. Quaisquer programas similares (parcialmente ou totalmente) terão **nota zero**, independente de qual for o original e qual for a cópia.

Programas desenvolvidos em outras linguagens (diferentes de C/C++) não serão aceitos.

3. Data e Forma de Entrega

Data de entrega: **28/09/2017** (5a-feira), até **23:59h**.

A data de entrega será considerada a data da última submissão dos arquivos no Tidia-ae.

A cada dia de **atraso** será **descontado 1 ponto** da nota final do trabalho. Serão considerados os dias consecutivos independentemente se é dia útil ou não.

Encerrado o prazo máximo de 10 dias de atraso, não serão aceitos mais trabalhos.

Forma de entrega:

A entrega será realizada no ambiente Tidia-ae na Atividade **Trabalho 1**.

O que entregar?

Você deve entregar os arquivos contendo **apenas o seu programa fonte e o relatório**.

A pasta completa do projeto deve ser compactada em um único arquivo (com extensão "zip").

O nome do arquivo compactado deve ser formado pela sigla "T1-" concatenada à 1ª letra do nome concatenada ao último sobrenome de um dos integrantes do grupo.

Exemplo: T1-esousa.

4. Descrição do Problema: a Festa de 10 anos de Formados...

Ano: 2020 ... Acabou!!!!! Formados!
10 anos depois.... Festa da turma da EngComp 2016...

Sua tarefa é: dado um mapa de cidades onde moram egressos da turma, as estradas que as ligam e a quantidade de egressos em cada cidade, determinar a melhor cidade para realizar a festa segundo os critérios abaixo. O objetivo é apresentar a melhor solução para cada um dos critérios.

Critério 1) A cidade escolhida dever ser aquela em que o menor número possível de egressos tenham que se deslocar, com a menor distância possível. Como você deve saber, muitos dos egressos da EngComp residem em cidades como São Paulo e Campinas, onde há mais oportunidades de trabalho nas áreas em que atuam. Poucos ficam em cidades menores como São Carlos ou Araraquara. Nesse critério, devemos considerar, além da distância de cada cidade a todas as outras, a quantidade de egressos que precisariam se deslocar para festa.

Critério 2) A cidade escolhida deve ser aquela "intermediária", considerando todos os caminhos mais curtos entre todas as cidades. Para tanto, pode ser usada uma medida chamada *betweenness centrality*. De modo geral, o valor de *betweenness* de um vértice é a proporção de caminhos mais curtos que passam por ele, considerando todos caminhos mais curtos de todos os partes de vértices do grafo. Um ponto importante é que o vértice em avaliação não pode ser nem origem e nem destino do caminho. Nesse critério, a cidade da festa da turma será aquela com o maior valor de *betweenness*.

4.1. ENTRADA

Para o cálculo da cidade mais adequada para festa, seu programa deve receber o número de cidades e estradas consideradas no mapa, a definição das estradas que ligam as cidades e a quantidade de egressos em cada uma delas.

Essas informações serão fornecidas no seguinte formato:

A primeira linha contém dois números inteiros **N** e **M**.

- **N** ($2 \leq \mathbf{N} \leq 50$) indica o número de cidades que serão consideradas no mapa. Isso indica que as cidades terão os códigos **0**, **1**, …, **N-1**; e,
- **M** ($1 \leq \mathbf{M} < \mathbf{N}^2$) representa a quantidade de estradas entre as cidades.

Em cada uma das próximas **N** linhas, seu programa deve receber um inteiro **E** ($0 \leq \mathbf{E} \leq 50$), que descreve o número de egresso naquela cidade. Em outras palavras, na primeira linha haverá o número de egressos que moram na cidade 0, na segunda, o número de egressos na cidade 1 e assim sucessivamente.

Nas **M** linhas seguintes, seu programa deve ler dois inteiros **ID1** ($0 \leq$ **ID1** < **N**) e **ID2** ($0 \leq$ **ID2** < **N**) e um número real **D** ($0 <$ **D** ≤ 100), indicando que há uma estrada de tamanho **D** que liga a cidade identificada por **ID1** àquela identificada por **ID2** (e vice-versa)

Exemplo

```
5 8
10
5
15
5
1
0 1 60.0
0 2 20.0
0 4 40.0
1 0 50.0
1 4 100.0
1 2 60.0
2 3 10.0
3 4 20.0
```

4.2. SAÍDA

Seu programa deve imprimir duas linhas, uma para cada critério descrito (na ordem em que eles foram apresentados). Para cada critério, a saída deve conter um inteiro identificando a cidade escolhida.

ATENÇÃO

- **Não** deverá ser utilizada qualquer variável global. Também, serão descontados pontos dos trabalhos que utilizam técnicas e estruturas de dados inadequadas ou ineficientes.

- **Não** poderão ser utilizadas bibliotecas com funções prontas (a não ser aquelas para entrada, saída e alocação dinâmica de memória).