

SME0230 - Introdução à Programação de Computadores

Primeiro semestre de 2017

Professora: Marina Andretta (andretta@icmc.usp.br)

Monitores: Douglas Buzzanello Tinoco (douglas.tinoco@usp.br)

Amanda Carrijo Viana Figur (amanda.figur@usp.br)

Exercícios de laboratório 10

Data: 08/06/2017.

Data máxima de entrega: 23/06/2017, até às 23h59min. Exercícios entregues fora do prazo não serão aceitos.

Forma de entrega: Os exercícios deverão ser entregues por e-mail para

exercicios.sme0230.2017@gmail.com

e o título do e-mail deverá ser IPC2017_Ex10. Cada exercício deve estar em um arquivo, chamado

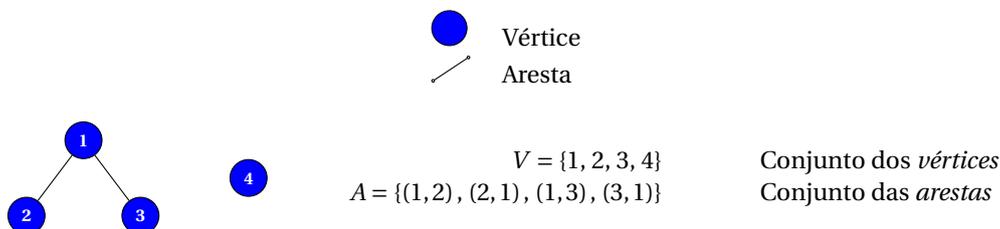
Ex10-<i>-IPC-<número usp>.c

com <i> o número do exercício e <número usp> o número USP do aluno.

No início do arquivo deve haver um comentário com o nome e o número USP do aluno.

Exercício 1

Um **grafo** é uma estrutura que consiste em um conjunto V de *vértices* e um conjunto A de *arestas*. Cada *aresta* é um par ordenado (v_1, v_2) com $v_1, v_2 \in V$ (ou seja, uma *aresta* é formada por um par de *vértices*). É mais fácil entender esse conceito através de um desenho:



Também podemos representar um grafo por uma matriz quadrada M , chamada **matriz de adjacências**, onde as colunas e as linhas representam os vértices.

Quando $M[i][j] = 1$ isso significa que os vértices i e j formam uma *aresta* (ou seja, estão conectados). Caso vértices não estejam conectados, eles recebem valor zero. Veja a **matriz de adjacências** do grafo acima:

	1	2	3	4
1	0	1	1	0
2	1	0	0	0
3	1	0	0	0
4	0	0	0	0

Chamamos de **grau do vértice** v o número de *arestas* que partem do *vértice* v . Segundo a **matriz de adjacências** de um grafo o **grau do vértice** v é a *soma dos valores da linha* v .

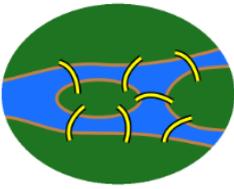
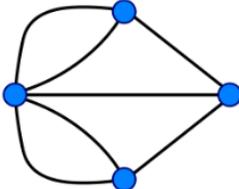
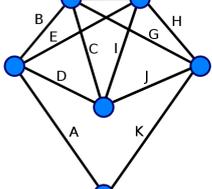
No nosso já famigerado exemplo temos os seguintes graus a partir da lista dos vértices:

1. grau 2;
2. grau 1;
3. grau 1;
4. grau 0.

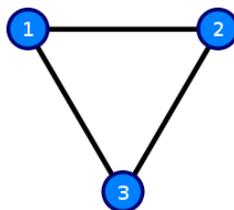
Definição 1. Um **circuito eureliano** é um caminho que percorre todas as arestas do grafo apenas uma vez, partindo do vértice v e terminando também no vértice v .

Dizemos que um grafo tem um **circuito eureliano** se, e somente se, cada vértice tem grau par.

O problema procurar e encontrar um **circuito eureliano** através de grafos foi proposto e resolvido por *Leonard Euler* em 1736 a partir do famoso *problema das sete pontes de Königsberg* (fonte: *Wikipédia*). Vejamos os seguintes exemplos:

		
<p>As pontes de Königsberg e sua representação via grafos, respectivamente. Cada "área verde" é um vértice e cada ponte as conectando é uma aresta do grafo. Note que nem todos os vértices têm grau par, portanto tal grafo não possui um circuito euleriano.</p>		<p>Cada vértice deste grafo tem grau par. Seguindo as arestas em ordem alfabética obtém-se um circuito euleriano.</p>

Implemente um algoritmo em linguagem C que receba um grafo através de uma **matriz de adjacências** e verifique se ele possui um **circuito euleriano**. Caso ele possua, verifique se todos os vértices do grafo possuem **grau** maior que zero. Caso a última afirmação seja verdadeira, receba do usuário um vértice válido do seu grafo e exiba um **circuito euleriano** a partir desse vértice. Por exemplo:



No grafo acima, partindo do vértice 1, podemos fazer o seguinte **circuito euleriano**:

- ```

1 2
2 3
3 1

```

Seu grafo deve ser representado por uma **matriz alocada dinamicamente!**

Note que os valores digitados podem não ser válidos: seu programa deve prever este caso.

Seu código **deve** estar **totalmente modularizado**. Isto é: todas as verificações descritas devem estar em funções.

**Não esqueça de fazer comentários!**

**Exercícios sem comentários claros ou indentados incorretamente terão nota ZERO!**