



SMAC03 – Grafos
Prof. Rafael Frinhani

ATIVIDADE 2 (AT2)

Assunto: 2. Teoria dos Grafos – Tipos e Representação – Matriz de Adjacências.

Data de Entrega: 28/08 até as 20:40h.

1. **Objetivo:** Verificar o aprendizado de conceitos básicos sobre grafos, implementar funções para operações em matrizes de adjacência.
2. **Descrição:** A atividade consiste na implementação de funções para operações em uma matriz de adjacências. Antes de realizar esta atividade é recomendado o estudo do conteúdo sobre o tópico “2. Teoria dos Grafos – Tipos e Representação” pelos slides da aula, complementando com as referências bibliográficas da disciplina. Em seguida, responda o Teste 1 relacionado a este conteúdo para verificação de conhecimentos teóricos. A atividade a ser executada é:

Implemente as funções a seguir:

- **tipoGrafo(matriz)**
Descrição: Retorna o tipo do grafo representado por uma dada matriz de adjacências.
Entrada: matriz de adjacências
Saída: Integer (0 – simples; 1 – dígrafo; 20 – multigrafo; 21 – multigrafo dirigido; 30 – pseudografo; 31 – pseudografo dirigido)
- **verificaAdjacencia(matriz, v_i , v_j)**
Descrição: Verifica se os vértices v_i e v_j são adjacentes.
Entrada: matriz de adjacências, v_i e v_j (ambos números inteiros que indica o id do vértice)
Saída: Boolean (True se os vértices são adjacentes; False caso contrário)
- **calcDensidade(matriz)**
Descrição: Retorna o valor da densidade do grafo.
Entrada: matriz de adjacências
Saída: Float (valor da densidade com precisão de três casas decimais)
- **insereAresta(matriz, v_i , v_j)**
Descrição: Insere uma aresta no grafo considerando o par de vértices v_i e v_j .
Entrada: matriz de adjacências, v_i e v_j (ambos são números inteiros que indicam o id do vértice)
Saída: matriz de adjacências (tipo numpy.ndarray) com a aresta inserida.
- **insereVertice(matriz, v_i)**
Descrição: Insere um vértice no grafo.
Entrada: matriz de adjacências, v_i (número inteiro que indica o id do vértice)
Saída: matriz de adjacências (tipo numpy.ndarray) com o vértice inserido.
- **removeAresta(matriz, v_i , v_j)**
Descrição: Remove uma aresta do grafo considerando o par de vértices v_i e v_j .
Entrada: matriz de adjacências, v_i e v_j (ambos são números inteiros que indicam os ids dos vértices)
Saída: matriz de adjacências (tipo numpy.ndarray) com a aresta removida.
- **removeVertice(matriz, v_i)**
Descrição: Remove um vértice do grafo.
Entrada: matriz de adjacências, v_i (número inteiro que indica o id do vértice)
Saída: matriz de adjacências (tipo numpy.ndarray) com o vértice removido.

3. **Entrega:** A entrega deverá ser feita exclusivamente pelo Moodle (e-mails não serão aceitos).

Observação: Na implementação siga fielmente a máscara da função (nome, parâmetros de entrada e de saída, tipos de dados).