

Atividade Prática A2 - Grafos

João Pedro Santana, 15204129

Gustavo Vicente Moser, 16204938

Stefano Bergamin Poletto, 16100745

Para facilitar o entendimento partes que estão presentes no código serão representadas com “”. Exemplo “vizinho = []”.

Outras funções utilizadas, “rotulo”, “vizinhos” e “ler” tem o mesmo funcionamento do trabalho A1 anterior.

Questão 1:

A questão tem como objetivo a criação de um programa que receba um arquivo indique os componentes fortemente conexos.

Para chegar no resultado necessário se cria um dicionário chamado “CTFA” onde nele se armazena se o vértice já foi visitado, pelo “c”, o tempo inicial de acesso ao vértice, pelo “t”, o tempo final, saída do vértice, pelo “f” e o “a” do vértice anterior.

É criado um dicionário chamado de “arestasT” que conterão as arestas transpostas e então será chamado a função dfs-adaptado com isso teremos os “CTFAALTERADO”, e então conseguiremos montar nossos conjuntos fortemente conexos, que serão construídos na função “conjuntoFortemente” que verifica os componentes que são fortemente conexos.

Nessa função criada se utiliza de duas listas “Q” e “conjuntosFortementes” após analisado o conjunto se retorna a lista “conjuntosFortementes”.

O restante do código segue como base o algoritmo disponibilizado na apostila, transformado para a linguagem Python.

Questão 2:

A questão tem como objetivo a criação de um programa que receba um arquivo e realize uma Ordenação Topológica.

Para chegar no resultado necessário se cria um dicionário chamado “CTF” onde nele se armazena se o vértice já foi visitado, pelo “c”, o tempo inicial de acesso ao vértice, pelo “t”, e o tempo final, saída do vértice, pelo “f”.

É criado a lista O que vai ser retornada para o usuário após a ordenação, para realizar essa ordenação se utilizou como base o algoritmo DFS e DFS-Visit-OT disponibilizado pelo professor na apostila de anotações, então esse algoritmo foi transformado para a linguagem de Python resultando no código anexado.

Questão 3:

A questão tem como objetivo a criação de um programa que receba um arquivo e realize o Algoritmo de Prim.

A qual foram criados dois dicionários “AK”, “Q”. O “AK” é composto por todos os vértices com suas características de “K” peso e “A” antecessor, e o “Q” com o valor dos vértices, que não foram percorridos, direcionando-o para o peso.

Utilizei da função min para pegar o menor peso existente no dicionário “Q” e quando encontrado este valor, é retirado do dicionário.