Universidade Federal de Santa Catarina

Departamento de Informática e Estatística

Ciência da Computação

INE5413 – Grafos

Professor: Rafael de Santiago

Equipe:

Anthony Bernardo Kamers (19204700)

Julien Hervot de Mattos Vaz (19202276)

Ricardo Jensen Eyng (19203166)

Relatório - A2

Florianópolis, 03 de fevereiro de 2022

Introdução:
Ao utilizar a biblioteca de grafos criada na Atividade 1, foram feitas algumas alterações pontuais, pois vimos que facilitava a implementação de alguns algoritmos. A modificação foi mudar o nome da classe *Node* para *Vertice* e, também, os atributos de *Vertice* *edgesEntrada* e *edgesSaida* são agora uma lista de *Arco* (classe utilizada tanto para arco quanto para arestas, onde grava de onde sai, para onde vai e o peso entre elas).

Questão 1:
Foi utilizada uma classe *VerticeDFS* para abstrair os vértices da busca em largura. Além disso, foi utilizada uma lista de *VerticeDFS* (variável local *vertices*), cada um contendo seu tempo final de busca e seu antecessor.
Na função *DFS_visit* foi utilizada uma lista dos vizinhos de cada vértice de origem. Por fim, são utilizadas duas listas para separar as raízes dos filhos, sendo essas listas de *VerticeDFS.*
Por fim, para printar no formato do exercício, foi preciso fazer uma lista para colocar o caminho de cada componente fortemente conexa (variável local *caminho*).

Questão 2:
Foi criada novamente uma classe auxiliar *VerticeDFS* para a busca em largura. Além disso, foi criada uma lista de *VerticeDF*S (variável local *vertices*), para fazer a busca da ordenação topológica. Foi utilizada implicitamente uma lista dos vizinhos de cada vértice na seguinte chamada: *verticeOrigem.vertice.vizinhosSaida()..*
Foi também utilizada uma lista para fazer o ordenamento topológico (variável local *O).* Primeiramente, essa variável é uma lista de *VerticeDFS*, depois a mapeamos para uma lista de strings, para printar no formato do exercício.

Questão 3:
Foram utilizadas três listas: *A*, *S* e *arestas*. A estrutura *arestas* é uma lista de *Arco* (servindo como aresta), que depois a ordenamos pelo peso. *S* é uma lista de inteiros com os *id*s dos vértices (classe Vertice). *A*, por sua vez, conterá a árvore geradora mínima, sendo esta uma lista de *Arco*.