

# Relatório - Atividade 2

Maria Eduarda de Melo Hang  
Thiago San't Helena Silva

Maio 2019

**Observação:** Cada questão contém um arquivo executável separado e as devidas instruções estão contidas neles. Os algoritmos utilizados para as resoluções das questões estão contidos no diretório *src* e no arquivo *functions.py*.

## 1 Questão 1

**Algoritmo:** O algoritmo implementado para essa questão é o de Componentes Fortemente Conexas (CFC) que se encontra em *src/functions.py* com a assinatura *cfc(G)*.

**Estruturas de Dados:** Um dicionário em Python foi utilizado para os vetores C, T, F e A para mapear cada vértice corretamente. Essa estrutura de dados foi utilizada para os métodos: *cfc(G)*, *dfsAdaptado(G, F)* e *dfsVisit(G, v, C, T, A, F, tempo)*.

**Saída:** Um pequeno tratamento é feito com o método *mostrarSubArvores(At, G)* para separar cada componente e depois as mostrar na tela.

## 2 Questão 2

**Algoritmo:** O algoritmo implementado nessa questão é o de Ordenação Topológica que se encontra em *src/functions.py* com a assinatura *ordenacaoTopologica(G)*.

**Estruturas de Dados:** As mesmas estruturas da Questão 1 para os vetores C, T, F e A foram utilizadas, a única diferença se encontra no uso da lista *O* para entregar a ordem topológica dos vértices.

**Saída:** Utiliza-se outra lista *Orotulos* para receber cada rótulo de um vértice encontrado na lista *O* e depois mostrá-la na tela.

### 3 Questão 3

**Algoritmo:** O algoritmo implementado nessa questão é o Kruskal que se encontra em *src/functions.py* com a assinatura *kruskal(G)*.

**Estruturas de Dados:** Utilizou-se, para o vetor *A*, um dicionário, o conjunto *S* como uma lista de conjuntos para cada vértice pertencente ao grafo *G*, uma lista para *E'* (representado como *E* no código) e outra lista nomeada *caminho* para guardar as arestas da árvore.

**Saída:** Para encontrar a soma da árvore, a lista *caminho* foi utilizada para agregar cada peso de uma aresta da árvore à variável *soma*. Essa mesma lista foi usada para imprimir todas as arestas.