Alunos: Eduardo Willwock Lussi (19200414) e Igor Perazzoli (19100529).

Adaptações para o grafo dirigido: O grafo dirigido utilizado é uma cópia do grafo não-dirigido com adaptações. Assim, esse grafo dirigido é composto de arcos, que possuem vértices de origem e destino e um peso. A maioria dos métodos sofreram breves alterações, mas a sua lógica é a mesma em comparação com o gráfico não-dirigido.

Questão 1: Para o algoritmo que encontra Componentes Fortemente Conexas (CFC), são utilizadas quatro listas: “C” para marcar os vértices já conhecidos/visitados, “T” para controlar os tempos de início das visitas, “F” para guardar os tempos de término das visitas e “A” para armazenar os ancestrais na árvore de busca em profundidade. Além disso, também é criado um outro grafo transposto do grafo passado como parâmetro da função, e deste são obtidas as listas “Ct”, “Tt”, “At” e “Ft”, assim como foi feito anteriormente com o grafo normal, para então utilizar a lista de ancestrais do grafo transposto para imprimir a solução ao usuário.

Questão 2: A implementação do algoritmo de ordenação topológica é composta de três funções. DFS é a função principal e itera sobre os vértices do grafo “G”, visitando-os através da função DFS_Visit_OT, que faz uma busca em profundidade e monta a ordenação topológica. Já a função imprimirDFS imprime a ordenação topológica conforme o padrão estabelecido no enunciado. As estruturas de dados utilizadas foram “C” e “O”, que são listas de vértices, “C” é a lista de vértices visitados, e “O”, a solução do problema, que é uma lista de vértices topologicamente ordenados.

Questão 3: O algoritmo escolhido para obter a árvore geradora mínima foi o Prim. No código são utilizadas uma lista “A”, para obter a ancestralidade dos vértices, e “K” mantém as chaves que definem qual o próximo vértice a ser selecionado para entrar na árvore. Outra estrutura útil nessa questão é “Q”, que nada mais é que uma cópia da lista de vértices que será usada para delimitar os vértices já incluídos na árvore da solução. Ademais, antes de iniciar o laço do while deve-se definir arbitrariamente uma posição de origem, isso é feito ao atribuir o valor “0” para uma posição da lista “K”, sendo que a posição escolhida deve ser maior ou igual que “0” e menor que a quantidade total de vértices do grafo.