

Relatório Atividade 2 Grafos - INE5413
Leonardo Luiz Gambalunga Alves de Oliveira (21201250)
Lucas Gusmão Valduga (21103505)
Ciências da Computação - UFSC

Para realizar esse trabalho escolhemos implementar os algoritmos em Python e utilizamos majoritariamente o material *Anotações para a Disciplina de Grafos (Versão de 27 de março de 2023, Rafael de Santiago)* como base para implementação dos algoritmos.

Componentes fortemente conexas:

A implementação do algoritmo de componente fortemente conexa foi baseado nos algoritmos 15, 16 e 17 da apostila. Além das listas que já existem nos pseudocódigos citados, ainda foram adicionados o Ct, como vértices transpostos visitados e Ca, como vértices visitados como resposta. Outro ponto importante é a criação do grafo transposto, cujo diferencial são as arestas invertidas, isso foi feito na representação do grafo. Para a ordenação da lista Fv, tempo final, pelo maior tempo, foi utilizado o algoritmo de quicksort e depois feito o for em ordem inversa.

Ordenação topológica:

Para implementar esse algoritmo foram utilizadas quatro estruturas principais, conforme o pseudocódigo que consta na apostila, C, T, F e O onde todas são vetores e armazenam respectivamente, (C) os vértices que já foram visitados, (T) o tempo de início do vértice na chamada dfsVisitOT, (F) o tempo final do vértice quando sua chamada dfsVisitOT é finalizada e (O) vetor da ordenação topológica propriamente dito que será o retorno final do algoritmo.

Prim:

Na implementação do algoritmo de Prim, foi utilizado o algoritmo 22 da apostila como referência, mas com algumas alterações. Dessa forma, foi alterado a variável Q, para uma lista Qv inicialmente com os valores False e conforme se visita respectivo vértice substitui o seu valor referente ao índice por True. Além disso, foi acrescentado outra lista, sendo o tamanho o dobro do tamanho de vértices, que garante que o caminho não se repita ou entre em um loop entre dois vértices, já que é um algoritmo que trata de grafos não-dirigidos, assim, quando é encontrando um caminho menor que Kv, é adicionado a aresta e o inverso dela. Por fim, outra diferença é o retorno, cujo neste algoritmo é retornada a soma dos pesos das arestas e a lista de Av que representa as arestas, sendo o índice do Av um dos vértices (com a soma 1) e o conteúdo o outro vértice.