



Universidade Estadual Vale do Acaraú – UVA

Ciências da Computação

Algoritmo para Grafos

Prof. Alex Martins

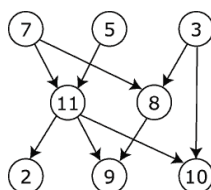
20/03/2023

Entrega: Até 26/03/2023



Lista de Avaliação 1(Para compor AP) (LA-1)

1. Teste de igualdade. Escreva uma função `GRAPHequal()` que decida se dois grafos, digamos G e H , são iguais.
2. Grafos não dirigidos. Escreva uma função `GRAPHremoveArc()` que receba dois vértices v e w de um grafo G representado por listas de adjacência e remova o arco v - w de G .
3. Grafo completo. Escreva uma função que construa um grafo completo com V vértices.
4. Torneio aleatório. Escreva uma função `GRAPHrandTournament()` que constua um torneio aleatório: para cada par v w de vértices distintos, exatamente um de v - w e w - v , com igual probabilidade, é um arco.
5. Verifica passeio. Escreva uma função booleana `GRAPHcheckWalk()` que verifique se uma dada sequência `seq[0..k]` de vértices de um grafo é um passeio. Faça duas versões da função: uma supõe que o grafo é dado por sua matriz de adjacências e outra supõe que o grafo é dado por suas listas de adjacência.
6. Faça uma lista de todos os ciclos simples no grafo definido pelo conjunto de arcos 0-2 1-0 2-1 2-3 3-0 3-1.
7. Exiba uma numeração topológica do grafo da figura. A numeração não precisa ser injetiva: dois ou mais vértices podem ter o mesmo número. Exiba também uma numeração anti-topológica, uma permutação topológica, e uma permutação anti-topológica.



8. Escreva uma função booleana eficiente `isRootedForest()` que decida se um grafo é uma floresta radcada.
9. Escreva uma função que receba uma árvore radcada e devolva a raiz da floresta.
10. Atualize suas bibliotecas. Acrescente as funções sugeridas nos exercícios acima à biblioteca `GRAPHmatrix`. Também acrescente as versões apropriadas à biblioteca `GRAPHlists`. Atualize os correspondentes arquivos-interface.