



Exercício de Fixação (EF-1)

1. ★ Faça uma figura bonita do grafo cujos arcos são indicados a seguir. Uma figura é tão mais bonita quanto mais simétrico e quanto menor o número de cruzamentos de linhas.

0-1 1-2 2-3 3-0 0-4 4-5 5-1 4-3 5-2

2. Certo ou errado? Num grafo não-dirigido, se $v-w$ é um arco então $w-v$ também é um arco. Num grafo (dirigido), se $v-w$ é um arco então $w-v$ não é um arco.
3. Escreva uma função que identifique os sorvedouros de um grafo. (Sugestão: calcule um vetor `isSink[]` tal que `isSink[v]` é `TRUE` se e somente se v é um sorvedouro.) Depois, escreva uma função que identifique as fontes de um grafo. (Sugestão: calcule um vetor `isSource[]` tal que `isSource[v]` é `TRUE` se e somente se v é uma fonte.)
4. [Sedgewick 17.40] Escreva uma função `GRAPHindeg()` que calcule o grau de entrada de um vértice v de um grafo G . Escreva uma função `GRAPHoutdeg()` que calcule o grau de saída de v .
5. Considere o problema de decidir se um vértice v é isolado num grafo G . Quanto tempo consome a solução do problema? Dê sua resposta em função do número de vértices do grafo.
6. Considere o problema de decidir se dois vértices são adjacentes num grafo G . Quanto tempo consome a solução do problema? Dê sua resposta em função do número de vértices e arcos do grafo.
7. Remoção de arco. 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 .
8. ☆ [Sedgewick 17.26] Considere o grafo definido pelos arcos abaixo. Faça uma figura do vetor de listas de adjacência quando os arcos são inseridas por `GRAPHinsertArc()`, na ordem dada abaixo, em um grafo inicialmente vazio.

3-7 1-4 7-8 0-5 5-2 3-8 2-9 0-6 4-9 2-6 6-4

9. É grafo não-dirigido? Escreva uma função `GRAPHisUndirected()` que decida se um dado grafo é não-dirigido.
10. Graus. Escreva uma função `UGRAPHdegrees()` que receba um grafo não-dirigido e devolve um vetor `grau[]`, indexado por vértices, tal que `grau[v]` é o grau do vértice v .

11. ☆ Prepare uma biblioteca de funções `GRAPHmatrix` para trabalhar com grafos representados por matriz de adjacências de acordo com o visto na aula. Atualize a biblioteca sempre que necessário.
12. ☆ Prepare uma biblioteca de funções `GRAPHlists` para trabalhar com grafos representados por listas de adjacência. Adapte as instruções do exercício anterior.
13. Instâncias extremas. Que acontece se `GRAPHrand()` for chamada com A maior que $V * (V-1)$? Que acontece se `GRAPHrand()` for chamada com $G \rightarrow V$ maior que `RAND_MAX+1`?
14. O grafo cujos arcos são 0-1 1-2 2-3 4-5 5-6 6-7 8-9 9-10 9-11 admite uma numeração topológica? Em caso afirmativo, o grafo é uma floresta radcada?
15. Escreva uma função booleana eficiente `isRootedForest()` que decida se um grafo é uma floresta radcada.
16. Todos os caminhos. Faça uma lista de todos os caminhos simples com exatamente 4 vértices no grafo definido pelos arcos 7-3 1-4 7-8 0-5 5-2 3-8 2-9 0-6 4-9 2-6 6-4.