lista de exercícios

1. Desenhe um grafo onde o resultado de uma busca em profundidade seja diferente do de uma busca em largura. Mostre seu resultado usando a representação “ingênua” de grafos.

2. Existe uma simplificação do algoritmo de coloração de grafos onde, ao invés de percorrermos o grafo em profundidade, colorimos os nós sequencialmente (nó 0, 1, ...).

   (a) Escreva esse algoritmo.

   (b) Mostre que ele não dá a colocação mínima para o grafo abaixo:

   ```
   (5 ((0 2) (0 3) (2 4) (3 4) (1 4)))
   ```

3. Escreva uma função que receba um grafo não dirigido na representação ‘ìngênua” e retorne o grafo representado por listas de adjacências. Lembre-se que, como o grafo é não dirigido, cada arco deve entrar nas listas de dois nós. Exemplo:

   ```
   > (converte '(4 ((0 1) (0 2) (2 3) (3 0))))
   #((1 2 3) (0) (0 3) (0 2))
   ```

4. Considere o algoritmo de árvore geradora de custo mínimo.

   (a) dado um grafo com n nós e m arcos, qual o número máximo de vezes que a função `loop` será chamada? Justifique.

   (b) Para terminar o loop, poderiamos testar se todos os nós já estão no conjunto `dentro`, ao invés de testarmos se `l` é vazia. Implemente essa modificação. Esses dois testes são equivalentes? Justifique.

   (c) Considere o grafo não dirigido abaixo:

   ```
   (4 ((0 1 1) (0 3 2) (0 2 5) (1 2 2) (3 2 1)))
   ```

   Qual será o valor de `l` na segunda vez que a função `loop` for chamada?

5. Qual o resultado do algoritmo de ciclos em grafos? Como “interpretá-lo” para sabermos se o grafo tem ciclos?

6. Escreva uma função que receba um grafo dirigido, representado por listas de adjacências, e retorne uma cópia “invertida” desse grafo (isso é, cada arco do novo grafo corresponde a um arco do grafo original com direção trocada).

7. Escreva um algoritmo que, dado um grafo dirigido e dois nós, retorne um caminho do primeiro para o segundo nó. Dica: extraia o caminho do resultado de uma busca em largura.