Algoritmos em Grafos
Lista de Exercícios III

1. Dado um grafo direcionado $G = (V, E)$ representado com uma matriz de adjacência, mostre como determinar se G possui um destino universal s em $O(V)$. Um vértice $s \in V$ é um destino universal se $d_{in}(s) = |V| - 1$ e $d_{out}(s) = 0$.

2. Mostre um exemplo onde a árvore resultante da busca em largura é diferente dependendo da ordem dos vértices na lista de adjacência.

3. Mostre que o valor distância $d[v]$ calculado na busca em largura independe da ordem dos vértices na lista de adjacência.

4. Projete um algoritmo $O(V + E)$ para determinar se um grafo não direcionado é bipartido ou não. Caso o grafo seja bipartido o seu algoritmo deve gerar a bipartição dos vértices.

5. O diâmetro de uma árvore é definido como

$$max_{u,v \in V} P^*_{uv}$$

ou seja o maior valor dentre os tamanhos de todos caminhos mínimos entre qualquer par de vértices da árvore. Mostre como computar em $O(V + E)$ este valor e prove a corretude do seu algoritmo.

6. Na DFS mostre que uma aresta (u, v) é
   - Uma aresta da árvore se e somente se $d[u] < d[v] < f[v] < f[u]$.
   - Uma aresta de retorno se e somente se $d[v] < d[u] < f[u] < f[v]$.
   - Uma aresta de cruzamento se e somente se $d[v] < f[v] < d[u] < f[u]$.

7. Suponha a conjectura: dado um digrafo G, suponha que na execução de DFS sobre G exista um caminho de u para v e que $d[u] < d[v]$, então v será um descendente de u. Mostre que a conjectura é falsa com um contra exemplo.

8. Mostre como é possível que na DFS um vértice u que tenha tanto arestas de saída quanto de entrada, pode terminar a DFS em uma árvore contendo somente ele próprio.

9. Mostre como alterar a DFS para reconhecer os componentes conexos de um grafo não direcionado. Se G possuir k componentes o algoritmo deve atribuir para cada $v \in V$ um valor $comp[v] \in [1, k]$ indicando a qual componente o vértice v pertence. Analise a corretude e complexidade do seu algoritmo.