Universidade Federal do Ceará
Departamento de Computação

Estruturas de Dados
Lista de exercícios 2

1. Faça um algoritmo que receba inteiros positivos N e K e monte uma árvore com N nós rotulados de 1 a N de modo que para todo $1 \leq i \leq N$, se o nó i tem filhos, então todos os nós de 1 a $i-1$ tem exatamente K filhos. Ou seja, construa uma árvore K-ária com N nós por níveis: o nó i só pode ter filhos se todos os anteriores a ele tiver K filhos.

2. Reescreva o algoritmo HeapSort usando uma estrutura de árvore binária, ao invés de um vetor.

3. Considere o problema de receber um grafo direcionado (representado por lista de adjacências) e decidir se ele representa ou não um grafo não-direcionado. Ou seja, para toda aresta xy, yx também é aresta? Faça um algoritmo com tempo $O(n+m)$ para resolver esse problema.

4. Faça um algoritmo que receba um grafo direcionado G (representado por lista de adjacências) e obtenha o grafo **transposto** de G, que é o grafo obtido invertendo a direção das arestas de G. Faça isso sem gerar a matriz de adjacências de nenhum dos dois grafos.

5. Dado um grafo direcionado G, o **quadrado** G^2 de G é o grafo direcionado tal que (u, v) é uma aresta de G^2 se e só se existe um vértice x tal que (u, x) e (x, v) são arestas de G. Faça um algoritmo para obter a lista de adjacências de G^2 a partir da lista de adjacências de G (sem usar matrizes de adjacência).

6. Dados dois grafos direcionados G_1 e G_2, o **produto cartesiano** $H = G_1 \square G_2$ é o grafo direcionado tal que, para todo vértice u_1 de G_1 e todo vértice u_2 de G_2, existe o vértice (u_1, u_2) de H. Além disso, existe uma aresta em H de (u_1, u_2) para (v_1, v_2) se e só se ($u_1 = v_1$ e u_2v_2 é uma aresta de G_2) ou ($u_2 = v_2$ e u_1v_1 é uma aresta de G_1). Faça um algoritmo para obter o produto cartesiano de dois grafos direcionados a partir de suas listas de adjacências.

7. Dizemos que um grafo direcionado é **acíclico** se não possui um ciclo direcionado (em outras palavras, todas as suas componentes fortes tem tamanho 1). Uma **ordenação topológica** de um grafo direcionado acíclico é uma ordenação linear de todos os seus vértices de modo que, para toda aresta (x, y), o vértice x aparece antes do vértice y nesta ordenação. Faça um algoritmo que, dado um grafo direcionado, determine se ele é acíclico e, caso seja, obtenha uma ordenação topológica dele. **Dica:** busca em profundidade.

8. Dizemos que um vértice de um grafo não-direcionado conexo é uma **articulação** se sua remoção desconecta o grafo. Faça um algoritmo de tempo $O(n+m)$ para obter todas as articulações de um grafo não-direcionado conexo. **Dica:** busca em profundidade.

9. Dizemos que uma aresta de um grafo não-direcionado conexo é uma **ponte** se sua remoção desconecta o grafo. Faça um algoritmo de tempo $O(n+m)$ para obter todas as pontes de um grafo não-direcionado conexo. **Dica:** busca em profundidade.

10. Reescreva os algoritmos de **Dijkstra** e de **Prim** considerando que os grafos são representados por matrizes de adjacências. Os algoritmos devem ter tempo $O(n^2)$ e não podem usar lista de adjacências.