UNIVERSIDADE FEDERAL DO CEARÁ - CAMPUS CRATEÚS
CIÊNCIA DA COMPUTAÇÃO E SISTEMAS DE INFORMAÇÃO

Aluno(a): _____ **Matrícula:** _____
CRT0390 - Algoritmos em grafos **Período: 2022.1**
Prof. Rennan Dantas

Nota: _____

2ª. ETAPA

Instruções para resolução da lista:
1 – A lista deve ser respondida de forma manuscrita, incluindo os grafos.
2 – Use preferencialmente caneta esferográfica de tinta azul ou preta para escrever as respostas. Certifique-se de que as suas respostas estão legíveis.
3 – Gere um PDF único com todas as suas respostas. Envie esse arquivo gerado pelo SIGAA.

1. Seja (u, v) uma aresta de peso mínimo em um grafo conexo G. Mostre que (u, v) pertence a alguma árvore geradora mínima de G.

2. Mostre que, se uma aresta (u, v) está contida em alguma árvore geradora mínima, então ela é uma aresta leve que cruza algum corte do grafo.

3. Seja e uma aresta de peso máximo em algum ciclo do grafo conexo $G = (V, E)$. Prove que existe uma árvore geradora mínima de $G' = (V, E - e)$ que também é uma árvore geradora mínima de G. Isto é, existe uma árvore geradora mínima de G que não inclui e.

4. Mostre que um grafo tem uma árvore geradora mínima única se, para todo corte do grafo, existe uma aresta leve única que cruza o corte. Mostre que a recíproca não é verdadeira, dando um contraexemplo.

5. O algoritmo de Kruskal pode devolver diferentes árvores geradoras para o mesmo grafo de entrada G, dependendo de como as ligações são rompidas quando as arestas são ordenadas. Mostre que, para cada árvore geradora mínima T de G, existe um modo de ordenar as arestas de G no algoritmo de Kruskal, de tal forma que o algoritmo retorne T.

6. Neste problema, apresentamos os pseudocódigos para três algoritmos diferentes na Figura 1. Cada um toma um grafo conexo e uma função peso como entrada e retorna um conjunto de arestas T. Para cada algoritmo, prove que T é uma árvore geradora mínima ou que T não é necessariamente uma árvore geradora mínima. Descreva também a implementação mais eficiente de cada algoritmo, quer ele calcule ou não uma árvore geradora mínima.

7. [POSCOMP - 2013 - Adaptado] Assinale a alternativa que apresenta, corretamente, o algoritmo utilizado para determinar o caminho mínimo entre todos os pares de vértices de um grafo.

   (a) Bellman-Ford
   (b) Floyd-Warshall
   (c) Dijkstra
   (d) Kruskal
   (e) Prim

   Além disso, explique resumidamente como cada um desses algoritmos funcionam e qual o objetivo de cada um.

8. O professor Borden propõe um novo algoritmo de divisão e conquista para calcular árvores geradoras mínimas, que apresentamos a seguir. Dado um grafo $G = (V, E)$, particione o conjunto V de vértices em dois conjuntos V_1 e V_2, tais que a diferença entre $|V_1|$ e $|V_2|$ seja no máximo 1. Seja E_1 o conjunto de arestas incidentes somente em vértices de V_1 e seja E_2 o conjunto de arestas incidentes somente em vértices de V_2.
   Resolva recursivamente um problema de árvore geradora mínima para cada um dos dois subgrafos $G_1 = (V_1, E_1)$ e $G_2 = (V_2, E_2)$. Finalmente, selecione a aresta de peso mínimo em E que cruza o corte (V_1, V_2) e use essa aresta para unir as duas árvores geradoras mínimas resultantes em uma única árvore geradora. Demonstre que o algoritmo calcula corretamente uma árvore geradora mínima de G ou dê um exemplo para o qual o algoritmo não funciona.

a. MAYBE-MST-A(G, w)

```
1  ordenar as arestas em ordem não crescente de pesos de arestas w
2  T = E
3  for cada aresta e, tomada em ordem não crescente de peso
4        if T − {e} é um grafo conexo
5              T = T − e
6  return T
```

b. MAYBE-MST-B(G, w)

```
1  T = ∅
2  for cada aresta e, tomada em ordem arbitrária
3        if T ∪ {e} não tem nenhum ciclo
4              T = T ∪ {e}
5  return T
```

c. MAYBE-MST-C(G, w)

```
1  T = ∅
2  for cada aresta e, tomada em ordem arbitrária
3        T = T ∪ {e}
4        if T tem um ciclo c
5              seja e' uma aresta de peso máximo em c
6              T = T − {e'}
7  return T
```

Figura 1: Fonte: Livro Algoritmos - Cormen

9. Prove ou refute: Seja $G = (V, E)$ um grafo dirigido ponderado com vértice fonte s e função peso $w : E \to \mathbb{R}$ e suponha que G não contenha nenhum ciclo negativo que possa ser alcançado de s. Então, para cada vértice $v \in V$, existe um caminho de s a v se e somente se BELLMAN-FORD termina com $v.d < \infty$ quando é executado em G.

10. Dê um exemplo simples de grafo dirigido com arestas de peso negativo para o qual o algoritmo de Dijkstra produz respostas incorretas. Explique o porquê.

11. Seja $G = (V, E)$ um grafo dirigido ponderado com vértice fonte s e suponha G inicializado por INITIALIZE-SINGLE-SOURCE(G, s). Prove que, se uma sequência de etapas de relaxamento define $s.p$ com um valor não NIL, então G contém um ciclo de peso negativo.

12. Seja $G = (V, E)$ um grafo dirigido ponderado com função peso $w : E \to \mathbb{R}$ sem ciclos de peso negativo. Seja $s \in V$ o vértice fonte e suponha G inicializado por INITIALIZE-SINGLE-SOURCE(G, s). Prove que, para todo vértice $v \in V_p$, existe um caminho de s a v em G_p e que essa propriedade é mantida como um invariante para qualquer sequência de relaxamentos.

13. Seja $G = (V, E)$ um grafo dirigido ponderado que não contém nenhum ciclo de peso negativo. Seja $s \in V$ o vértice fonte e suponha G inicializado por INITIALIZE-SINGLE-SOURCE(G, s). Prove que existe uma sequência de $|V|–1$ etapas de relaxamento que produz $v.d = d(s, v)$ para todo $v \in V$.