Algoritmos em Grafos
Lista de Exercícios VII
Caminhos mínimos entre todos os pares

1. No algoritmo baseado em multiplicação de matrizes, mostre que a operação de multiplicação definida pelo algoritmo é associativa.

2. Mostre como expressar o problema de caminhos mínimos de uma única fonte s como um produto de vetores e matrizes. Mostre como o computo desta forma corresponde ao que é feito pelo algoritmo Bellman-Ford.

3. Suponha que desejamos computar uma matriz de predecessores Π dos caminhos mínimos de tal forma que Π_{ij} indica o predecessor de j no caminho mínimo de i para j. Suponha que já temos computado uma matriz L com as distâncias dos caminhos mínimos entre todos os pares de vértices. Mostre como computar Π a partir de L em $O(n^3)$.

4. Mostre como alterar o algoritmo baseado em multiplicação de matrizes para computar as matrizes de predecessores $\Pi^1, \ldots, \Pi^{(n-1)}$, tal que numa matriz Π^m_{ij} indica o predecessor de j no caminho mínimo de i para j com até m arestas.

5. Modifique os algoritmos baseados em multiplicação de matrizes de tal forma que estes usem apenas 2 matrizes durante toda sua execução (fora a matriz W).

6. Mostre como usar o algoritmo baseado em multiplicação de matrizes para detectar a existência de um ciclo negativo no grafo.

7. Modifique o algoritmo de Floyd-Warshal para que este compute matrizes de predecessor Π^k, onde Π^k_{ij} indica o predecessor de j no caminho mínimo de i para j que usa apenas vértices $1, 2, \ldots, k$ como vértices intermediários.

8. No algoritmo original de Floyd-Warshal usamos $n+1$ matrizes $D^0, D^1, \ldots, D^n$ que indicam os caminhos mínimos restritos aos i primeiros vértices, como vértices intermediários no caminho, para $i = 0, 1, 2, \ldots, n$. Desta forma o algoritmo usa espaço $O(n^3)$. Mostre que o algoritmo abaixo é correto e usa apenas $O(n^2)$ de espaço.

Function `floyd-warshall`(D):

$n \leftarrow W.rows$;
$D \leftarrow W$;
for $k \leftarrow 1$ **to** n **do**
  for $i \leftarrow 1$ **to** n **do**
    for $j \leftarrow 1$ **to** n **do**
      $D_{ij} \leftarrow \min(D_{ij}, D_{ik} + D_{kj})$;
    end
  end
end
return D;

9. Projete um algoritmo $O(VE)$ para computar o fecho transitivo de um digrafo G.

10. Suponha que para se encontrar o fecho transitivo em um grafo direcionado **acíclico** $G = (V, E)$, tenhamos custo $f(V, E)$ (f é uma função de custo sobre o tamanho do grafo). Dado um grafo direcionado qualquer $G = (V, E)$, mostre como computar o seu fecho transitivo $G^* = (V, E^*)$ em tempo $f(V, E) + O(V + E^*)$.

11. Suponha um digrafo G onde $w(u, v) \geq 0$ para todo $(u, v) \in E$. No algoritmo de Johnson computamos $h(v)$ para todo $v \in G$ e setamos $\hat{w}(u, v) = w(u, v) + h(u) - h(v)$. Qual será o valor de $\hat{w}(u, v)$ para grafos conforme o enunciado?

12. No algoritmo de Johnson podemos deixar todas as arestas de G' com peso não negativo alterando-se os pesos da seguinte forma: ache a aresta de menor peso
$$w^* = \min_{(u,v) \in E} w(u, v).$$
Agora, para cada aresta (u, v) setamos $\hat{w}(u, v) = w(u, v) - w^*$, de tal forma que todas as arestas agora possuem pesos não negativos. O algoritmo funciona corretamente ao se usar esta nova função de pesos?

13. Considere o algoritmo de Johnson sobre um digrafo G, onde criamos um novo vértice s e o conectamos com todos os demais vértices de G com arestas de custo 0, obtendo um novo digrafo G'. Depois é executado o algoritmo de Bellman-Ford sobre G'. Suponha agora que ao invés deste processo, apenas selecionamos um vértice s qualquer de G e aplicamos o algoritmo de Bellman-Ford diretamente em G com este s. Mostre que este novo processo pode levar o algoritmo de Johnson a computar uma solução errada. Mostre ainda que se G for fortemente conexo então o algoritmo de Johnson funciona corretamente com esta alteração.