1. OBSERVAÇÃO: como discutido em sala, para aplicarmos a técnica da programação dinâmica, é necessário que o problema em questão possua a propriedade da "subestrutura ótima", que pode ser explicada como segue. Considere uma instância "I" de um problema "P", e uma solução ótima "S" para "I". Considere ainda que "S" é uma composição de soluções S_1 ... S_k para instâncias I_1 ... I_k, estas últimas também relativas ao problema "P" e obtidas por uma decomposição de "I". Nesse caso, "P" possui a propriedade da subestrutura ótima se cada solução S_i é necessariamente uma solução ótima para a correspondente instância I_i.

   Assim, por exemplo, os problemas de caminhos mínimos em grafos discutidos em sala possuem a propriedade da subestrutura ótima, mas o problema do caminho mais longo não.

   Referências: - Cormen, §15.3
                - http://en.wikipedia.org/wiki/Optimal_substructure

2. CAMINHOS MÍNIMOS ENTRE TODOS OS VÉRTICES EM GRAFOS PONDERADOS: o problema é encontrar, para cada par de vértices (u,v) de um grafo direcionado simples de arestas ponderadas por números reais e sem circuitos de peso <= 0, um caminho de peso mínimo de "u" a "v". O grafo de entrada é fornecido por meio de uma matriz W tal que

   ```
   W[i,j] = 0,                se i = j;
   W[i,j] = o peso de (i,j), se (i,j) é aresta do grafo;
   W[i,j] = INFINITO,        se i != j e (i,j) não é aresta do grafo.
   ```

   (Observação: o conjunto de vértices do grafo é {1 ... n}.)

   A saída do algoritmo, na versão simplificada, é uma matriz n x n D tal que D[i,j] é o peso de um caminho mínimo do vértice "i" ao vértice "j", ou então INFINITO, se nenhum tal caminho existe.
   Na versão completa do problema, é necessário retornar também uma matriz P tal que P[i,j] é o predecessor de j num caminho mínimo de i a j, ou então NULO, se i = j ou se i != j e não há caminho de i a j.

3. EXERCÍCIO: como discutido em sala, a seguinte é uma estratégia de programação dinâmica para resolver o problema em questão:

   ```
   01. PARA k DE 1 A n
   02. | Compute os caminhos mínimos entre todos os pares de vértices da parcela
       | do grafo de entrada que contém apenas os vértices de 1 a k (ou, mais
       | formalmente, todos os caminhos mínimos do subgrafo induzido pelo
       | conjunto de vértices {1 ... k}).
   ```

   Escreva então um algoritmo O(n^3) que materializa a estratégia acima.
   DICA: como discutido em sala, a ideia é:

   a) Ao fim da iteração k, temos os caminhos mínimos entre os vértices 1 ... k.

   b) Calcule então, para cada i de 1 a k, um caminho mínimo de k+1 a i. Para cada "i", isso custa O(n), com base nos dados da iteração "k", pois há O(n) possibilidades para o segundo vértice do caminho.

   c) Analogamente, para cada i de 1 a k, calcule um caminho mínimo de i a k+1. Como acima, isso leva O(n^2) no total (para todos os valores de "i").

   d) Agora, para cada par (i,j) tal que i != j e i,j ∈ {1 ... k}, calcule um caminho mínimo de i a j na parte do grafo que contém os vértices de 1 a "k+1". Claramente, um tal caminho mínimo ou é um caminho mínimo que não passa por "k+1", e portanto que já conhecemos da iteração "k", ou então é um caminho que passa por "k+1", e portanto formado por um caminho mínimo de "i" a "k+1" e um caminho mínimo de "k+1" a "j", que já foram computados nos passos "b" e "c" acima.

4. EXERCÍCIO: o algoritmo do exercício anterior funciona se o grafo puder possuir ciclos de peso zero? Se não, é possível modificá-lo de forma que sim? Além disso, estenda o algoritmo de forma a resolver a versão completa do problema.

5. ALGORITMO DE FLOYD-WARSHALL:

```
================================================================================
Algoritmo: FloydWarshall.
Entrada: a matriz W n x n descrita acima.
Saída: Uma matriz D n x n.
--------------------------------------------------------------------------------
PARA i DE 1 A n
| PARA j DE 1 A n
| | D[i,j] := W[i,j]
PARA k DE 1 A n
| PARA i DE 1 A n
| | PARA j DE 1 A n
| | | D[i,j] := mín { D[i,j] , D[i,k] + D[k,j] }
RETORNE D.
================================================================================
```

6. EXERCÍCIOS:

a) Resolva a versão completa do problema, isto é, retorne não apenas o valor de uma solução ótima, mas também a solução em si. Suponha que o grafo pode possuir ciclos de peso zero.

b) Diga como é possível detectar a existência de circuitos de peso negativo por meio do algoritmo de Floyd-Warshall.

c) Resolva o seguinte problema relacionado em O(n^3): dado um grafo direcionado G, calcule o fecho transitivo de G, é o grafo que possui uma aresta (u,v) se e somente se há um caminho de u para v em G.