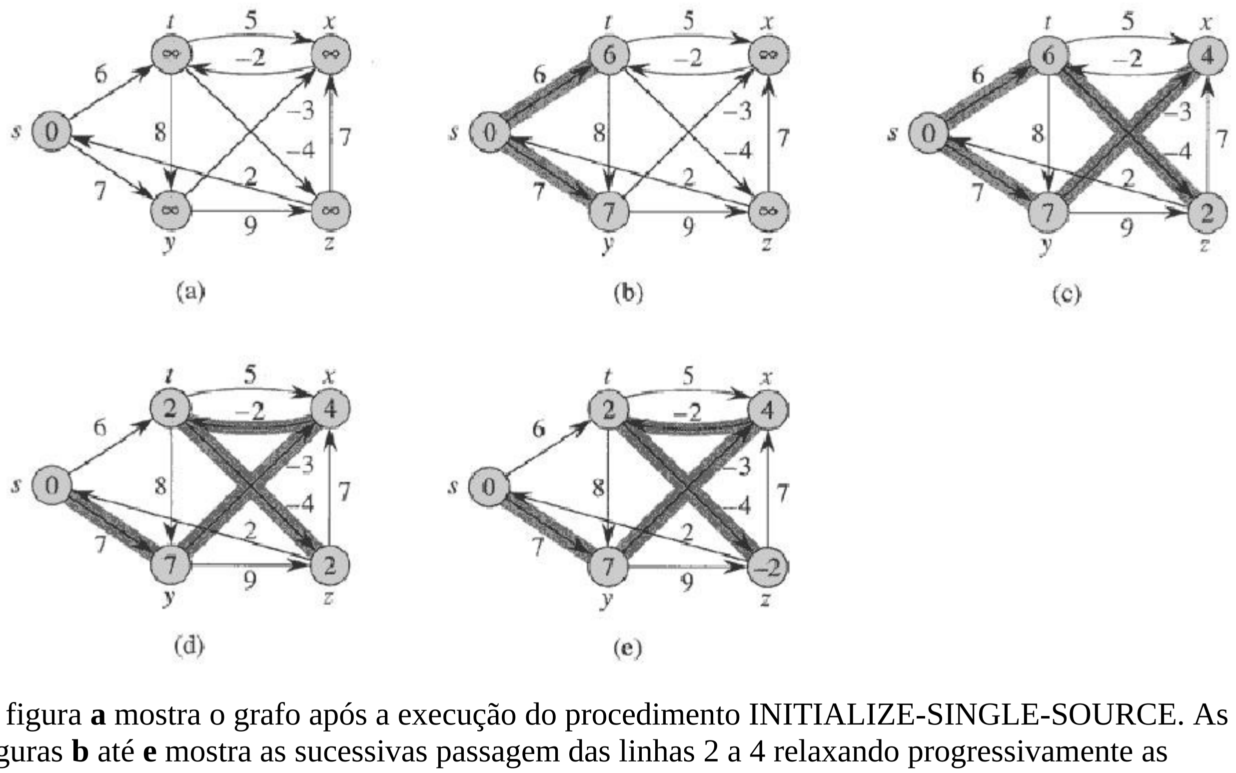
O algoritmo de Bellman-Ford (Fábio):

O algoritmo de Bellman-Ford resolve o problema do caminho mais curto de única origem para o caso mais geral, no qual os pesos das arestas podem ser negativos. Dado um grafo orientado ponderado *G = (V, E)* com origem em s e função peso *w:E →* **R**, o algoritmo retorna FALSE quando encontra um ciclo de peso negativo indicando que não existe solução, ou retorna TRUE indicando que produziu os caminhos mais curtos e seus pesos. O algoritmo usa a técnica do relaxamento, diminuindo progressivamente a estimativa d[v] no peso de um caminho mais curto.

```
BELLMAN-FORD(G, w, s)
1  INITIALIZE-SINGLE-SOURCE(G, s)
2  for i ← 1 to |V[G]| − 1
3       do for each edge (u, v) ∈ E[G]
4              do RELAX(u, v, w)
5  for each edge (u, v) ∈ E[G]
6       do if d[v] > d[u] + w(u, v)
7             then return FALSE
8  return TRUE
```

O algoritmo começa iniciando as estimativas d[v] nos pesos dos caminhos mais curtos com *infinito* e os predecessores π [v] com NIL, através do procedimento INITIALIZE-SINGLE-SOURCE. Nas linhas 2 a 4 tenta relaxar todas as arestas de todos os vértices. E finalmente verifica a existência de um ciclo de peso negativo nas linhas de 5 a 7.

A figura a seguir mostra a execução do algoritmo de Bellman-Ford sobre um grafo de 5 vértices.



A figura **a** mostra o grafo após a execução do procedimento INITIALIZE-SINGLE-SOURCE. As figuras **b** até **e** mostra as sucessivas passagem das linhas 2 a 4 relaxando progressivamente as arestas. Na figura **e** temos os valores finais para d e π . Nesse exemplo o algoritmo retorna TRUE, pois não há ciclos de pesos negativos.

O **algoritmo Bellman-Ford é executado no tempo *O(V E).*** A inicialização através de INITIALIZE-SINGLE-SOURCE demora Θ *(V)*, cada uma das passagens sobre as arestas nas linhas 2 a 4 demora o tempo Θ (*E)*, e o ciclo para verificar ciclos de peso negativo das linhas 5 a 7 demora o tempo de Θ *(E)*.

Um pouco mais sobre caminhos:

Durante o transcorrer da aula em vários tópicos a expressão ***caminho*** causou dúvidas sobre a sua definição. Prof. Meidanis informou que as definições usuais são:

Um **caminho** de comprimento k de um vértice u até um vértice u' em um grafo G = (V, E), é uma seqüência < $v_{0}, v_{1}, ..., v_{k}$ > de vértices tais que u = v_{0} , u' = v_{k} e ($v_{i-1},$, v_{i}) para i = 1,2,...k, onde todos os vértices do caminho são distintos.

Um **passeio** é um caminho onde vértices e arestas podem ser repetidas.

Uma **trilha** é um caminho onde apenas os vértices podem ser repetidos.

As definições apresentadas pelo livro texto (no Apêndice C pag. 854), são diferentes. O livro chama de **caminho simples** o caminho usual, e de caminho o que é chamado de passeio. Todo o conteúdo do livro toma as definições do apêndice como base.

Referências bibliográficas:

(livro texto) Cormen, T.H.;Leiserson, C.E.; Rivest, R.L.; Stein, C.; Algoritmos. Tradução da 2ª edição americana Teoria e Prática, 2002.