Construção e Análise de Algoritmos

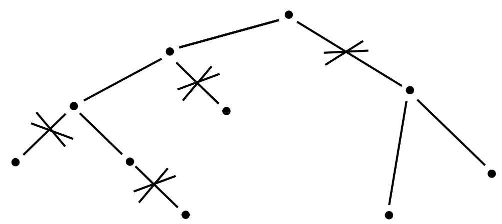
lista de exercícios 26

1. **Desconectando uma árvore**

   Considere uma árvore (i.e., um grafo conexo sem ciclos) onde um dos vértices foi designado como a raiz, e cada aresta tem um peso associado a ela.

   O problema consiste em remover um subconjunto de arestas da árvore de modo que a raiz fique desconectada de todas as folhas da árvore.

   A figura abaixo apresenta uma solução simples para esse problema



   O objetivo consiste em encontrar um conjunto de arestas que desconecta a árvore com peso total mínimo.

   a) Apresente um algoritmo de programação dinâmica para esse problema.

   b) Estime a complexidade do seu algoritmo

2. **Blocos de montar**

   Imagine que você tem blocos de montar de vários tipos

$$b_1, \; b_2, \; b_3, \; \ldots, \; b_n$$

   Cada bloco b_i possui um comprimento x_i, uma largura y_i e uma altura z_i, mas você pode rotacioná-lo e usar qualquer lado como base.

   Você possui vários blocos de cada tipo, e o problema consiste em empilhar os blocos para construir uma torre da maior altura possível.

   A única restrição é que, ao colocar um bloco em cima do outro, a base do bloco de cima deve ser estritamente menor do que a base do bloco de baixo.

   a) Apresente um algoritmo de programação dinâmica para esse problema.

   b) Estime a complexidade do seu algoritmo

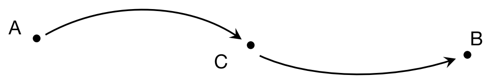
3. **Caminhos mais longos**

Nas aulas 17 e 26, nós vimos algoritmos que encontrarm caminhos mais curtos em um grafo direcionado.

Mas, o problema análogo de encontrar um caminho mais longo entre dois vértices é bem mais difícil.

A razão é a seguinte.

Quando nós temos um caminho mais curto de A até B que passa por um vértice intermediário C



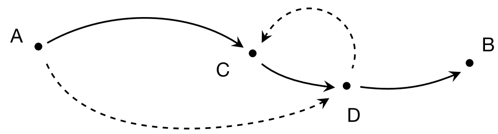
o trecho desse caminho entre A e C (e entre C e B também) é um caminho mais curto entre os dois vértices.

Dessa maneira, o problema de encontrar caminhos mais curtos entre vértices distantes se decompõe naturalmente em problemas mais simples de encontrar caminhos mais curtos entre vértices próximos.

No entanto, a situação não é assim quando nós consideramos caminhos mais longos.

Quer dizer, se o caminho na figura acima é um caminho mais longo de A até B, então não há garantia de que o trecho entre A e C seja um caminho mais longo de A até C.

Isso é o caso porque o caminho mais longo de A até C poderia passar por um vértice D no trecho entre C e B



E quando isso acontece, nós não podemos concatenar o maior caminho de A até C com o trecho entre C e B do caminho original (que poderia ser o caminho mais longo entre esses dois vértices), para obter um caminho mais longo de A até B.

Essa concatenação não é um caminho válido porque ela passa duas vezes pelo vértice D.

Mas, note que essa situação só é possível porque o grafo tem um ciclo: a volta que a concatenação dá em torno do vértice D.

A boa notícia é que em grafos sem ciclos a programação dinâmica é capaz de resolver o problema dos caminhos mais longos.

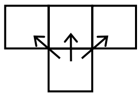
a) Apresente um algoritmo de programação dinâmica que encontra caminhos mais longos de um vértice v até todos os outros que ele pode alcançar, em um grafo direcionado acíclico G com distâncias associadas às arestas.

Dica: Em um grafo direcionado sem ciclos, sempre é possível encontrar uma ordem $v_1, v_2, \ldots, v_n$ para os vértices do grafo de modo que todas as arestas apontam para o lado direito (i.e., levam de um vértice v_i a um vértice v_j com $i < j$).

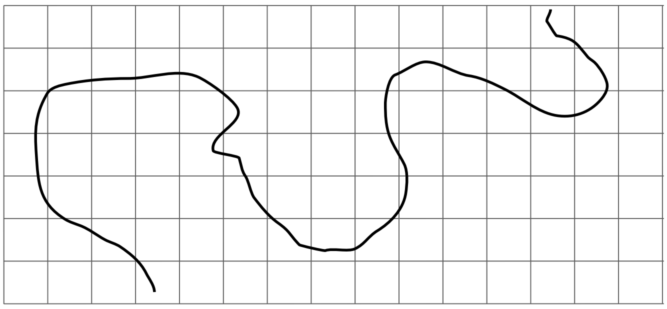
b) Estime a complexidade do seu algoritmo.

4. **Expedição nas montanhas 2**

No problema 3 da lista 24, nós introduzimos a seguinte restrição para simplificar as coisas: "*os expedicionários decidiram que só vão considerar trajetórias que fazem algum progresso a cada passo*"



Mas, em uma situação real, é bem possível que o caminho mais fácil para atravessar a cadeia de montanhas seja um caminho bem tortuoso



Portanto, dessa vez, nós vamos resolver o problema sem essa restrição.

Quer dizer, lembre que nós temos um mapa `A[n,m]` que indica a altitude de cada quadradinho dentro da região.

E o objetivo consiste em encontrar um caminho que leva de um lado ao outro da cadeia de montanhas, com o menor deslocamento vertical possível (i.e., a soma das subidas e descidas a cada passo).

Como antes, a trajetória pode começar em qualquer ponto na parte de baixo do mapa, e terminar em qualquer ponto na parte de cima.

a) Apresente um algoritmo de programação dinâmica para esse problema.

b) Estime a complexidade do seu algoritmo

5. **Algoritmo de Viterbi**

(. . .)