MC448 – Projeto e Análise de Algoritmos
Lista de Exercícios 7

Orlando Lee

Os números referem-se a segunda edição de CLRS.

1. (CLRS 15.3-3) Considere uma variante do problema de multiplicar uma cadeia de matrizes na qual o objetivo é parentizar a seqüência de matrizes de modo a **maximizar**, em vez de minimizar, o número de multiplicações. Este problema possui subestrutura ótima?

2. (CLRS 15.3-5) Em programação dinâmica primeiro resolvemos os subproblemas e então decidimos qual deles usar para obter uma solução ótima do problema. Professora Capuleto afirma que não é sempre necessário resolver todos os subproblemas de modo a obter uma solução ótima. Ela sugere que uma solução ótima do problema de multiplicar uma cadeia de matrizes pode ser sempre encontrada escolhendo a matriz A_k para dividir o subproblema de multiplicar $A_iA_{i+1}\cdots A_j$ escolhendo k que minimiza $p[i-1]p[k]p[j]$ **antes** mesmo de resolver os subproblemas. Mostre uma instância (exemplo) do problema de multiplicar uma cadeia de matrizes para a qual esta estratégia gulosa não resulta em uma solução ótima.

3. (CLRS 15.4-3) Descreva uma versão memorizada de LCS-Length com complexidade de tempo $O(mn)$.

4. (CLRS 15.4-5) Descreva um algoritmo de complexidade $O(n^2)$ para encontrar uma subseqüência crescente mais longa de uma seqüência de n números. Por exemplo, uma subseqüência crescente mais longa de $10, 5, 15, 7, 9, 21, 12$ é $5, 7, 9, 12$.

5. (não se assuste com o enunciado!) Considere que temos uma rede de computadores representada por uma matriz $R[1..n, 1..m]$. Cada posição da matriz representa um nó da rede (um computador), mas os valores das entradas de R são irrelevantes neste problema.

   - Um nó (i,j) (com $i < n$ e $j < m$) pode transmitir uma mensagem apenas para o nó $(i+1,j)$ ou para o nó $(i,j+1)$.
   - Um nó (n,j) (com $j < m$) pode transmitir uma mensagem apenas para o nó $(n,j+1)$.
   - Um nó (i,m) (com $i < n$) pode transmitir uma mensagem apenas para o nó $(i+1,m)$.
   - O nó (n,m) pode apenas receber mensagens (de $(n-1,m)$ ou de $(n,m-1)$).

   Há um custo de transmissão sempre que um nó transmite uma mensagem. O custo de transmitir uma mensagem de um nó (i,j) a um nó (i',j') é denotado por $c(i,j,i',j')$ (naturalmente, (i',j') deve ser um dos nó aptos a receber a mensagem de (i,j) como descrito acima). Todos os custos são não-negativos.

   O problema consiste em encontrar uma rota para transmitir uma mensagem de $(1,1)$ até (n,m) de modo a minimizar o custo total de transmissão. Por exemplo, considere uma rede R com $n = 2$ e $m = 3$ onde $c(1,1,1,2) = 1, c(1,1,2,1) = 2, c(1,2,1,3) = 2, c(1,2,2,2) = 2, c(2,1,2,2) = 3, c(2,2,2,3) = 1$ e $c(1,3,2,3) = 1$. A rota com custo de transmissão mínimo é $(1,1), (1,2), (2,2), (2,3)$ com custo total 4.

   (a) Mostre que o problema tem subestrutura ótima. Descreva então uma recorrência para $T(i,j)$, o menor custo de uma rota de transmissão para uma mensagem de (i,j) a (n,m). Note que $T(n,m) = 0$.

   (b) Escreva um algoritmo baseado em programação dinâmica para encontrar o custo de transmissão mínimo e também a rota de transmissão de custo mínimo. Qual a complexidade do seu algoritmo?

6. (Subset-sum ou mochila simplificada) Escreva um algoritmo de programação dinâmica que resolve o seguinte problema: dados números inteiros não-negativos $w_1, \ldots, w_n$ e W, encontrar um subconjunto K de $\{1, \ldots, n\}$ que satisfaça $\sum_{k \in K} w_k \leq W$ e maximize $\sum_{k \in K} w_k$. Imagine que $w_1, \ldots, w_n$ são os tamanhos de arquivos digitais que você deseja armazenar em um disquete de capacidade W. Como se resolve o Problema da Partição: dados $w_1, \ldots, w_n$, determinar se existe um subconjunto $I \subset \{1, \ldots, n\}$ tal que $\sum_{i \in I} w_i = \sum_{i \in N-I} w_i$ onde $N = \{1, \ldots, n\}$?

7. (O enunciado é longo, mas é realmente simples de entender...) Um ex-aluno da UNICAMP gostaria de desenvolver um novo e sofisticado vídeo-gravador digital chamado ViTo. No software ViTo, um programa de televisão i é representado por uma tripla (c_i, s_i, t_i) onde c_i é o **canal do programa**, s_i é o horário de **início do programa** e t_i é o horário de **término do programa**.

   O proprietário do ViTo fornece uma lista de n programas de televisão que ele tem interesse em assistir e a cada programa i, ele atribui um **grau de interesse** r_i. ViTo pode gravar apenas um programa de cada vez e os horários de alguns programas podem ser **conflitantes**: dois programas i e j são conflitantes se $s_i \leq s_j < t_i$ ou $s_j \leq s_i < t_j$. Note que i e j **não** são conflitantes se $t_i = s_j$ (significando que j começa imediatamente depois de i).

   ViTo deve escolher um subconjunto dos programas a ser gravados de modo a **maximizar** o **grau de interesse agregado** dos programas escolhidos (ou seja, a soma dos graus de interesse dos programas escolhidos).

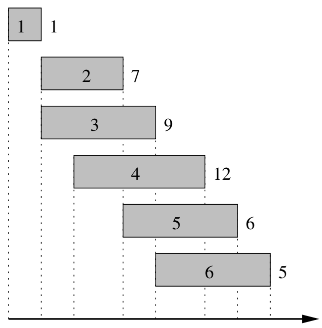

Figura 1: Exemplo de instância para o ViTo. Cada retângulo representa um programa (o canal foi omitido) e os números à direita deles representam o grau de interesse do usuário. Uma solução é gravar os programas $1, 2$ e 5 com graus de interesse $1, 7, 6$ totalizando 14. A melhor solução é gravar os programas $1, 3$ e 6 com graus de interesse $1, 9, 5$ totalizando 15.

   (a) Mostre que o problema tem subestrutura ótima. Denote por $T(n)$ o grau de interesse agregado de uma solução ótima para um conjunto de n programas. Escreva uma recorrência que define $T(n)$. **Sugestão:** considere uma solução ótima S do problema. Se o programa x com maior instante de término faz parte de S, então $S - \{x\}$ é solução ótima de qual subproblema? E se x não fizer parte de S?

   (b) Suponha que os programas estejam ordenados pelo valor de $t(i)$ em ordem não decrescente e suponha que seja dado um vetor $f[\,]$ tal que $f[i] = \max\{j : j < iet(j) \leq s(i)\}$. Escreva um algoritmo baseado em programação dinâmica que devolve um **conjunto de programas** com grau de interesse agregado máximo em tempo $O(n)$. Justifique a correção e a complexidade do seu algoritmo.