Análise e Técnicas de Algoritmos
Lista 6 de Exercícios - Período 2003.1

Aluno(a):

Para os problemas de 1 a 5 faça:

- Defina um algoritmo baseado em programação dinâmica que resolve o problema.
- Indique a sua complexidade.
- Defina uma instância do problema e aplique a sua solução.

1. Sejam u e v dois strings. Nós queremos transformar u em v com um menor número de operações possível. Existem apenas três tipos de operações: remover um caractere, adicionar um caractere e modificar um caractere.

2. Sejam n itens de tamanhos s_1, s_2, ..., s_n. Existe um subconjunto desses itens com a soma dos tamanhos exatamente igual a S? Defina um algoritmo baseado em programação dinâmica que resolva este problema. Analise a sua complexidade.

3. Sejam duas seqüências X e Y, uma seqüência Z é uma subseqüência comum de X e Y se ela é subseqüência tanto de x como de Y. O problema da maior subseqüência comum consiste em determinar o tamanho da maior subseqüência comum das seqüências X e Y.

4. O triângulo de Pascal é um trângulo de números em que cada elemento é a soma dos dois elementos acima, com exceção dos elementos das *pontas* que têm valor 1. Veja um exemplo de triângulo de Pascal para $n = 4$.

```
        1
      1   1
    1   2   1
  1   3   3   1
1   4   6   4   1
```

5. Considere uma seqüência de n inteiros positivos $x_1, x_2, ..., x_n$, alternados por um operador aritmético de soma (+) ou de multiplicação ($\times$), num total de $n - 1$ operadores $op_1, op_2, ..., op_{n-1}$. Essa seqüência corresponde

a uma expressão $EXP = x_1 op_1 x_2 op_2 ... op_{n-1} x_n$. Dependendo da ordem em que as operações são efetuadas, diferentes valores de EXP podem ser obtidos. O problema consiste em retornar o maior valor de EXP.

6. Sobre programação dinâmica:

   (a) Apresente sua definição.

   (b) Quais as suas vantagens?

   (c) Quais as suas desvantagens?

7. Considere o problema da multiplicação de n matrizes retangulares: $M = M_1.M_2...M_n$, onde cada M_i tem r_{i-1} linhas e r_i colunas. Se multiplicarmos uma matriz $p \times q$ por uma matriz $q \times s$, teremos que executar $p.q.s$ operações. Dependendo da ordem em que são efetuadas as operações de multiplicação, realizamos uma quantidade diferente de operações. Uma solução ótima para este problema é definir a ordem em que devemos proceder as multiplicações, fazendo um número mínimo de operações.

   O seguinte algoritmo apresenta uma solução por divisão-e-conquista para este problema. $custo(i,j)$ computa o custo mínimo para a multiplicação $M_i.M_{i+1}...M_j$.

   $custo(i,j)$
   if $i = j$ **then**
   return(0)
   else
   return($min_{i \leq k < j}(custo(i,k) + custo(k+1,j) + r_{i-1}.r_k.r_j)$)
   end if

   Transforme este algoritmo em uma solução que usa programação dinâmica.