Tópicos Avançados em Algoritmos - Folha de exercícios (revisão para o teste III)

Um ou mais destes exercícios (ou alíneas) serão incluídos na prova.

1. **Verdadeiro ou falso?**

   Diga se cada uma das seguintes afirmações é verdadeira ou falsa (não é necessário justificar)

   (a) Definição de Programação Dinâmica: técnica em que se utiliza uma programação "bottom-up", determinando soluções óptimas de sub-problemas, começando pelos de menor tamanho.

   (b) Usando convenientemente a Programação Dinâmica conseguem-se por vezes algoritmos polinomiais para problemas completos em NP (como o problema de decisão da mochila).

   (c) O número de comparações efectuadas pelo algoritmo de "merge" de 2 vectores com m e n elementos está necessariamente compreendido entre $m+1$ e $m+n-1$.

   (d) Existe uma constante inteira c, tal que qualquer algoritmo de ordenação baseado em comparações (de um vector com n elementos) efectua, para n suficientemente grande, pelo menos $cn \log n$ comparações no caso mais favorável.

2. **LCS: a mais longa sub-sequência comum**

   Considere a implementação estudada de um algoritmo para a maior sub-sequência comum às sequências x e y de comprimentos m e n, respectivamente. Seja $c(i,j)$ o tamanho da maior sub-sequência comum a $x[1,2,\ldots,i]$ e $y[1,2,\ldots,j]$.

   (a) Sendo $i \geq 2$ e $j \geq 2$, demonstre a seguinte recorrência para a determinação de $c(i,j)$.

   $$c(i,j) = \begin{cases} 1 + c(i-1, j-1) & \text{se } x_i = y_j \\ \max(c(i-1,j), c(i,j-1)) & \text{se } x_i \neq y_j \end{cases}$$

   Nota. Para cada caso deve mostrar que existe uma sequência máxima com o comprimento indicado (pode haver mais que 1 sequência com o comprimento máximo).

   (b) Descreva em pseudo-código um algoritmo directamente derivado da recorrência anterior.
   Nota. Este algoritmo é exponencial.

   (c) Descreva em pseudo-código o algoritmo derivado da recorrência, *usando tabelação.* Não se esqueça de considerar devidamente os casos base ($m = 0$ e $n = 0$)
   Nota. Este algoritmo é polinomial.

3. **Procura de x**

Considere o problema de procurar um valor x num vector `v[1..n]` (não necessariamente ordenado) com n elementos distintos. A resposta é o índice `i` tal que `v[i]`$= x$, ou `-1` se x não ocorre em `v`. Utiliza-se o modelo externo dos dados com a análise no pior caso; todos os acessos aos dados da forma "$x =$`v[i]`?". Seja $c(n)$ o número de acessos aos dados (comparações) no pior caso.

(a) Usando o Princípio da Informação necessária, determine um minorante de $c(n)$.

(b) Determine o majorante seguinte: $c(n) \leq n$.

(c) Mostre que se verifica o minorante[1] $c(n) \geq n$.

(d) Mostre que se usarmos a pesquisa binária (supondo que o vector está ordenado), temos um número de comparações $c(n)$ que satisfaz $c(n) \leq \log(n+1)$, onde se supõe que n é da forma $n = 2^p - 1$. Que conclui neste caso (pesquisa binária) do minorante obtido na alínea 3a?

4. **Comparações no "merge"**

Considere o algoritmo de `merge(u[1..m],v[1..n])` onde todos os $m+n$ elementos dos vectores são distintos. Usa-se o modelo externo dos dados, com a análise no pior caso; todos os acessos aos dados da forma "$x =$`u[i]<v[j]`?".

(a) Mostre que cada resultado possível pode ser representado por uma sequência de $m+n$ caracteres `u` ou `v`, com m caracteres `u` e n caracteres v e <u>vice-versa</u>.
Um exemplo (com $m = 4$ e $n = 5$): `merge([1,2,8,10],[3,4,9])` $\Rightarrow$ `uuvvuvu`

(b) Usando o Princípio da Informação Necessária determine um minorante do número de comparações efectuadas entre elementos de `u` e elementos de `v`.

(c) Supondo que $m = n$ e usando a fórmula de Stirling, $n! \approx \sqrt{2\pi n}(n/e)^n$, determine uma forma fechada (sem usar factoriais nem combinações) para o minorante obtido na alínea anterior.

5. **Ordenações...**

Seja `v[1..n]` um vector com n elementos distintos que se pretende ordenar. Mostre que qualquer algoritmo de ordenação baseado em comparações pode, para cada n, ter que efectuar (isto é, efectua no pior caso) $c(n) = \lceil \log(n!) \rceil$ comparações. Tabele esse minorante $c(n)$ do número de comparações para $n = 2, 3, 5, 6$.

[1] Este minorante é, portanto, exacto. Assim, o minorante obtido na alínea 3a é muito fraco.