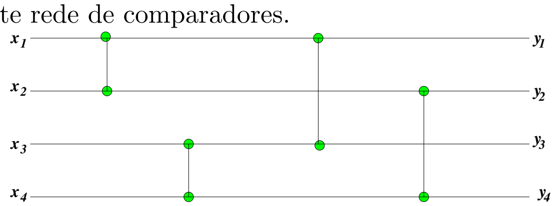
Tópicos Avançados em Algoritmos - época normal

Enunciado com frente e verso / Em cada alínea indica-se a cotação / Duração da prova: 2h15m

1. **16%** Diga se cada uma das seguintes afirmações é verdadeira ou falsa (não é necessário justificar)
   (a) É possível multiplicar 2 matrizes quadradas de dimensão $n \times n$ em tempo $O(n^{2.9})$
   (b) É possível multiplicar 2 matrizes quadradas de dimensão $n \times n$ em tempo $\Omega(n)$
   (c) É possível somar 2 matrizes quadradas de dimensão $n \times n$ em tempo $O(n\sqrt{n})$
   (d) RP $\subseteq$ NP
   (e) Se os valores a ordenar pertencerem a $\{1, 2, \ldots, u\}$ com u fixo, o tempo de execução do algoritmo de ordenação por contagem é de ordem $O(n)$.
   (f) Nenhum algoritmo (determinístico) que determine a mediana de um vector pode ter uma ordem de grandeza $O(\sqrt{n})$.

2. Considere o problema de determinar o elemento de ordem i de um vector `v[1..n]`.
   (a) **9%** Foi estudado um algoritmo (baseado na ideia do "quick sort") para este problema com uma eficiência $O(n)$ em termos do tempo médio. Descreva esse algoritmo numa linguagem informal, de forma sucinta e clara.
   (b) **4%** "Existe um outro algoritmo para o mesmo problema que garante uma eficiência $O(n)$ no pior caso". A afirmação anterior é verdadeira ou falsa? Justifique.

3. Considere a seguinte rede de comparadores.



   (a) **4%** Indique os sucessivos valores nas linhas da rede quando as linhas de entrada contêm os valores $x_1 = 8$, $x_2 = 5$, $x_3 = 3$, $x_4 = 1$.
   (b) **7%** Mostre que a rede não é uma rede de ordenação.
   (c) **7%** Defina formalmente *profundidade* de uma rede de comparadores.

4. Custo amortizado do contador binário
   (a) **4%** Defina custo amortizado de uma sequência de operações.
   (b) **9%** Considere um contador binário, inicialmente com 0; são efectuados n operações de `INCR` (incremento de 1) e pretende-se mostrar que 2 é um custo amortizado dessa sequência de operações. O modelo de custos reais é o seguinte: cada bit que muda de valor contribui com 1 para o custo. Por exemplo, o custo de `INCR(0101011)`→`0101100` é 3. Pode (mas não é obrigado a) usar um método directo em que se contabiliza para cada $i \geq 0$ o número de vezes que o bit de ordem i é modificado.

5. Programação Dinâmica
   (a) **4%** Defina o método genérico conhecido como Programação Dinâmica.
   (b) **7%** Relativamente ao método recursivo directo, a Programação Dinâmica permite muitas vezes substanciais melhorias de eficiência. Explique porquê.

Continua do outro lado

6. LCS, a mais longa sub-sequência comum (elementos não necessariamente consecutivos). 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) $\boxed{\textbf{9\%}}$ Sendo $i \geq 1$ e $j \geq 1$, demonstre a seguinte recorrência para a determinação de $c_{i,j}$.

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

   (b) $\boxed{\textbf{7\%}}$ Represente esquematicamente o algoritmo derivado da recorrência, *usando tabelação.* Não se esqueça de considerar devidamente os casos base ($m = 0$ ou $n = 0$)
   Nota. Este algoritmo (com tabelação) é polinomial, de ordem $O(mn)$.

7. "Hash" perfeito

   (a) $\boxed{\textbf{4\%}}$ Defina "hash" perfeito.

   (b) $\boxed{\textbf{9\%}}$ Suponha que é já conhecido o algoritmo aleatorizado para a construção do "hash" perfeito com tabela de "hash" de tamanho n^2. Descreva em linhas gerais o algoritmo para a construção do "hash" perfeito com tabela de "hash" de tamanho $O(n)$.