MAC 5711 – Análise de Algoritmos
PRIMEIRO SEMESTRE DE 1999
Primeira Prova – 26 de março de 1999

1. **(Valor: 2.0 pontos)** Decida (sem dar justificativa) se $f(n) = O(g(n))$, $f(n) = \Theta(g(n))$, $f(n) = \Omega(g(n))$ ou nenhuma destas alternativas, para
   (a) $f(n) = \log^k n$, onde $k \geq 1$ é uma constante, e $g(n) = n^\epsilon$, onde $\epsilon > 0$ é uma constante.
   (b) $f(n) = c^n$, onde $c > 1$ é uma constante, e $g(n) = n^k$, onde $k \geq 1$ é uma constante.
   (c) $f(n) = \log_a n$ e $g(n) = \log_b n$, onde $a, b > 1$ são constantes.
   (d) Escolha um dos itens acima e justifique a sua resposta.

2. Considere o seguinte algoritmo que determina o maior e o menor elemento de um vetor $V[p..q]$, com $n = q - p + 1 = 2^k \geq 2$ elementos.

```
maxmin(V, p, q, max, min)
  se (q-p=1)
    se (V[p] > V[q])                 (*)
      max = V[p];  min = V[q]
    senao
      max = V[q];  min = V[p]
  senao {
    m = (p+q)/2
    maxmin(V, p, m, max1, min1)
    maxmin(V, m+1, q, max2, min2)
    se (max1 > max2)                 (*)
      max = max1
    senao
      max = max2
    se (min1 < min2)                 (*)
      min = min1
    senao
      min = min2
  }
```

   (a) **(Valor: 1.0 ponto)** Escreva uma recorrência para o número de comparações (em função de $n = q - p + 1$) entre elementos do vetor (as comparações marcadas com *) que o procedimento acima executa.
   (b) **(Valor: 1.5 pontos)** Resolva a recorrência do item acima **exatamente** (lembre-se que estamos considerando apenas $n = 2^k \geq 2$), concluindo depois a ordem de grandeza do número de comparações do algoritmo (ou seja, conclua que o número de comparações é $O(g(n))$ ou $\Theta(g(n))$, para $g(n)$ tão simples quanto possível). Seja tão preciso quanto possível.

3. **(Valor: 1.5 pontos)** Mostre que qualquer algoritmo baseado em comparações, para decidir se n elementos são todos iguais, necessita de pelo menos $n - 1$ comparações.

4. **(Valor: 2.0 pontos)** Considere que a entrada é um conjunto S contendo n números reais, e um número real x.
   (a) Descreva um algoritmo que determine se existem dois elementos de S cuja soma é exatamente x. O seu algoritmo deve executar em tempo $O(n \log n)$. Justifique a sua resposta.
   (b) Suponha agora que o conjunto S dado está ordenado. Descreva um algoritmo para resolver o problema em tempo $O(n)$. Justifique a sua resposta.

5. **(Valor: 2.0 pontos)** Descreva um algoritmo o mais eficiente possível para determinar se dois conjuntos são disjuntos. Determine a complexidade do seu algoritmo em função do tamanho m e n dos conjuntos. Lembre-se de considerar, na sua análise, o caso em que m é muito menor que n.