1o. Exercício Escolar de
ALGORITMOS E ESTRUTURAS DE DADOS
13 de abril de 2000

1. *(Vale 2.0 pontos)*

   a) Diga se é possível ter $f = \Theta(g)$, se a função f tiver **sempre** o valor maior do que a função g.

   b) Explique o papel de cada uma das constantes envolvidas na definição de $\Omega(g)$.

   IMPORTANTE: Use no máximo 5 linhas para cada um dos itens.

2. a) *(Vale 1.5 pontos)*
   Desenhe a seqüência de árvores que se obtém pela construção de um *heap*, **usando a abordagem bottom-up**, a partir de um array com as seguintes chaves: 100, 250, 15, 141, 418, 79, 164, 503, 601.

   b) *(Vale 1.5 pontos)*
   Diga qual o custo de construir um heap bottom-up (como acima) **para um array qualquer de tamanho** n. Justifique a sua resposta.

3. *(Vale 3.0 pontos)*
   Escreva um algoritmo **linear** para identificar, **para cada nó da árvore**, quantos nós existem na sua sub-árvore à esquerda e quantos nós existem na sua sub-árvore à direita. Esta informação deve ser colocada em dois campos, *size-esq* e *size-dir*, dentro da estrutura do nó. Em seguida, indique o custo do algoritmo, justificando a cada linha.

   ENTRADA: Raiz r de uma árvore binária T (construída com apontadores).

   IMPORTANTE:
   Escreva o algoritmo em pseudo-código, ou seja, uma linguagem estruturada, para que o custo da computação seja facilmente identificado.

4. *(Vale 2.0 pontos)* Construa a árvore de decisão do algoritmo *Mergesort* com $n = 4$, até o nível cinco, inclusive.
   Obs: A raiz tem nível zero.