Curso de Tecnologia em Sistemas de Computação
Disciplina: Estrutura de Dados e Algoritmos
AP3 - Segundo Semestre de 2005

Nome -
Assinatura -

Observações:

1. Prova sem consulta e sem uso de máquina de calcular.
2. Use caneta para preencher o seu nome e assinar nas folhas de questões e nas folhas de respostas.
3. Você pode usar lápis para responder as questões.
4. Ao final da prova devolva as folhas de questões e as de respostas.
5. Todas as respostas devem ser transcritas nas folhas de respostas. As respostas nas folhas de questões não serão corrigidas.

---

1. (1,5) Falso ou verdadeiro? (Justifique): A complexidade de melhor caso de um algoritmo para um certo problema é necessariamente maior do que qualquer limite inferior para o problema.

   Resposta:

   Falso. Um limite inferior para um problema, por definição, é apenas um limite inferior para a *complexidade de pior caso*, e não para a de melhor caso. Portanto, é perfeitamente possível que a complexidade de melhor caso seja assintoticamente menor que um limite inferior para um problema.

2. (1,5) Falso ou verdadeiro? (Justifique): Uma árvore binária pode ser construída, de forma única, a partir do seu percurso em *pré-ordem* e a informação do *número de nós em cada subárvore*.

   Resposta:

   Verdadeiro. Suponha que o percurso em pré-ordem é dado pela sequência de nós $x_1 x_2 \dots x_n$. Sabemos que a raiz da árvore é o nó x_1, e que x_2 é um dos filhos da raiz. Seja $n(x_i)$ o número de nós na subárvore de raiz x_i, que é conhecido. Se $n(x_2) = j$, deduzimos que $x_2 \dots x_{1+j}$ são os nós que formam parte da subárvore T enraizada em x_2. Além do mais, $x_{2+j} \dots n$ (se existirem) são os nós que formam parte da subárvore T' enraizada em x_{2+j}, o outro filho da raiz. Aplicando recursivamente o mesmo raciocínio às subárvores T e T', conseguimos reconstruir a árvore binária original de forma única.

3. Dê exemplos de cadeias X e Y, de comprimentos n e m, respectivamente, nos seguintes casos:

   a. (1,5) O algoritmo de força bruta para casamento de cadeias efetua um número *máximo* de comparações. (Justifique)

      Resposta:

      O número máximo de comparações que o algoritmo de força bruta realiza é $m(n - m + 1)$, que corresponde a comparar todos os m caracteres da cadeia Y a cada iteração. Cada iteração corresponde a um caractere de X onde se inicia a comparação com Y. Há portanto um total de $n - m + 1$ iterações. Por exemplo, para $n = 9$ e $m = 5$, as cadeias poderiam ser:

      $X = AAAAAAAAA \qquad Y = AAAAB$

b. (1,5) O algoritmo de força bruta para casamento de cadeias efetua um número *mínimo* de comparações. (Justifique)

O número mínimo de comparações que o algoritmo de força bruta realiza é $n-m+1$, que corresponde a comparar apenas o primeiro caractere da cadeia Y a cada iteração. Cada iteração corresponde a um caractere de X onde se inicia a comparação com Y. Há portanto um total de $n - m + 1$ iterações. Por exemplo, para $n = 9$ e $m = 5$, as cadeias poderiam ser:

$X = AAAAAAAAA \qquad Y = BBBBB$

4. (2,0) Escreva um algoritmo recursivo para encontrar o maior elemento de uma lista. Determine sua complexidade.

Resposta:

Assumimos que desejamos encontrar o maior elemento de uma lista L entre os extremos inf e sup. Se a lista L tem $n \geq 1$ elementos, a chamada externa será $MAIOR(L, 1, n)$.

função $MAIOR(L, inf, sup)$

se $inf = sup$

    então $MAIOR := L[inf]$

    senão

        $m := \lfloor (inf + sup)/2 \rfloor$

        $a := MAIOR(L, inf, m)$

        $b := MAIOR(L, m + 1, sup)$

        $MAIOR := maximo(a, b)$

A complexidade do algoritmo acima é $O(n)$.

5. (2,0) Escreva o algoritmo de conversão de árvores graduadas em rubro-negras. Dê um exemplo de árvore graduada com altura pelo menos 4 (incluindo os nós externos), e mostre como é a árvore rubro-negra resultante da conversão.

Resposta:

Seja T uma árvore graduada. Atribui-se uma bicoloração a T percorrendo-se a árvore de cima para baixo. Se v é a raiz de T, então fazemos $cor(v) := negra$. Caso contrário, seja w o pai de v. Define-se $cor(v) := rubra$ se $posto(v) = posto(w)$, e $cor(v) := negra$ no caso contrário.

Um exemplo de conversão pode ser visto no livro-texto "Estruturas de Dados e seus Algoritmos". A árvore rubro-negra da página 147 (Figura 5.10) é o resultado da conversão da árvore graduada da página 146 (Figura 5.9b).