Curso de Tecnologia em Sistemas de Computação
Disciplina: Estrutura de Dados e Algoritmos
Gabarito da AP1 - Primeiro Semestre de 2010

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. Leia atentamente o texto a seguir, e depois resolva as questões abaixo.

   Sabemos que um problema computacional pode admitir vários algoritmos para resolvê-lo, que podem ter complexidades de pior caso diferentes. Dentre esses algoritmos, aqueles que possuem as menores complexidades de pior caso podem ou não ser ótimos, pois o que caracteriza um algoritmo ótimo não é o fato de ele ser "melhor" do que os outros, mas é algo relacionado à complexidade intrínseca do problema que ele resolve, isto é, ao limite inferior do problema.

   (a) (1,0) Dê a definição formal de limite inferior de um problema computacional.

   Resposta: O limite inferior de um problema P é uma função ℓ tal que a complexidade de pior caso de qualquer algoritmo que resolva P é $\Omega(\ell)$.

   (b) (1,0) Responda: Seja A um algoritmo ótimo para o problema computacional P. Então, qualquer outro algoritmo para P tem complexidade de pior caso estritamente maior (em termos assintóticos) do que a de A. Falso ou verdadeiro? Justifique.

   Resposta: Falso. Se A é um algoritmo ótimo para P, então A tem a menor complexidade de pior caso possível. Logo, qualquer outro algoritmo para P tem complexidade de pior caso maior ou igual à de A.

2. Considere uma biblioteca pública com um grande acervo de livros, no qual as seguintes operações são executadas:

   A. Busca de um livro no acervo.
   B. Inserção de um novo livro no acervo.
   C. Remoção de um livro do acervo.

   Suponha ainda que, para cada livro, existem as seguintes informações: título, autor e contador de buscas (número de buscas do livro, sempre atualizado a cada nova busca).

   Temos a seguir várias estruturas de dados que poderiam ser utilizadas para implementar o acervo de livros:
   1. Lista linear não ordenada.
   2. Lista linear ordenada por título.
   3. Lista linear ordenada por autor.
   4. Lista linear ordenada (crescentemente) por contador.
   5. Lista linear ordenada (decrescentemente) por contador.
   6. Lista encadeada não ordenada.
   7. Lista encadeada ordenada por título.
   8. Lista encadeada ordenada por autor.
   9. Lista encadeada ordenada (crescentemente) por contador.
   10. Lista encadeada ordenada (decrescentemente) por contador.

Resolva as questões a seguir (não há necessidade de justificar, apenas escreva as opções que se aplicam em cada caso):

(a) (1,0) Quais as melhores estruturas em relação à operação A?

Resposta: São as estruturas 2, 3 (porque são $O(\log n)$), 4, 5 e 10 (são $O(n)$, mas têm um bom caso médio).

(b) (1,0) Quais as melhores estruturas em relação à operação B?

Resposta: São as estruturas 1, 5, 6, 7, 8, 9 e 10 (desconsiderando a busca, todas admitem inserção em tempo $O(1)$).

(c) (1,0) Quais as melhores estruturas em relação à operação C?

Resposta: São as estruturas 1, 6, 7, 8, 9 e 10 (desconsiderando a busca, todas admitem remoção em tempo $O(1)$).

3. É dado um vetor V com $n \geq 1$ elementos não necessariamente distintos, isto é, *pode haver elementos repetidos*.

   Resolva as questões abaixo.

   (a) (1,0) Escreva um algoritmo que monta dois outros vetores W e R com as seguintes propriedades: o vetor W contém os mesmos elementos de V, mas eliminando as repetições, e o vetor R contém o número de ocorrências dos elementos de V, da seguinte forma: se $W[\,i\,]$ contém o elemento x de V, então $R[\,i\,]$ é igual ao número de vezes em que x ocorre em V. Exemplo: Se $V = [\ 2\ 4\ 6\ 2\ 5\ 1\ 1\ 1\ 6\ 7\ ]$, então $W = [\ 2\ 4\ 6\ 5\ 1\ 7\ ]$ e $R = [\ 2\ 1\ 2\ 1\ 3\ 1\ ]$. Observação: na resposta do seu algoritmo, a ordem dos elementos em W pode ser qualquer uma.

   Resposta:

```
W[1] := V[1]
R[1] := 1
ultimo := 1
para i := 2 até n faça
    achou := F
    j := 1
    enquanto j ≤ ultimo faça
        se W[j] = V[i] então
            R[j] := R[j] + 1
            achou := V
            j := ultimo + 1
        senão
            j := j + 1
    se achou = F então
        ultimo := ultimo + 1
        W[ultimo] := V[i]
        R[ultimo] := 1
```

(b) (1,0) Suponha que os vetores W e R já estejam calculados. Escreva um algoritmo que reordena os elementos de W em ordem decrescente do número de ocorrências em V. Para o exemplo acima, poderíamos ter como resposta $W = [\ 1\ 6\ 2\ 7\ 5\ 4\ ]$. (Outras ordens são possíveis, pois note que há empates; neste caso, qualquer uma delas é válida.) Observação: use os valores do vetor R para resolver este item.

Resposta: Seja m o número de elementos de W e de R.

```
para i := 1 até m − 1 faça
    maior := i
    para j := i + 1 até m faça
        se R[maior] < R[j] então
            maior := j
    se maior ≠ i então
        temp := W[maior]
        W[maior] := W[i]
        W[i] := temp
        temp := R[maior]
        R[maior] := R[i]
        R[i] := temp
```

4. (1,0) Considere uma pilha P contendo 5 posições de 1 a 5. A variável *topo* marca a posição do topo da pilha. No início, a fila P encontra-se vazia, e a variável *topo* vale zero.

   Usamos a notação R para denotar a operação de remoção de um elemento da pilha P, e a notação $I(X)$ para denotar a operação de inserção de um elemento X na pilha P.

   Considere a seguinte sequência de operações em P:

   $$I(A), I(B), I(C), R, I(D), R, I(E), I(G), I(H), R, R, R, R, I(J)$$

   Desenhe como fica a fila P após a sequência de operações acima, e forneça o valor final da variável *topo*. Use um traço (–) para denotar as posições vazias. Como um exemplo de configuração, poderíamos ter como resposta: $P = [\ C\ D\ H\ -\ -\ ]$, onde *topo* neste caso vale 3.

   Resposta: $P = [\ A\ J\ -\ -\ -\ ]$, $topo = 2$.

5. Resolva as questões a seguir.

   (a) (1,0) O número de subárvores vazias de uma árvore binária com n nós é sempre o mesmo, independentemente do formato da árvore. Verdadeiro ou falso? (Justifique)

   Resposta: Verdadeiro. Uma árvore binária com n nós possui exatamente $n + 1$ subárvores vazias.

(b) (1,0) Desenhe uma árvore binária *não completa* com 10 nós e a menor altura possível.

Resposta: