UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE
IMD0039 – ESTRUTURAS DE DADOS BÁSICAS II
PROFª SÍLVIA MARIA DINIZ MONTEIRO MAIA

LISTA DE EXERCÍCIOS – 2ª UNIDADE

1. Que outras representações existem para árvores além da hierárquica? Explique.
2. Prove que toda árvore com número de nós maior que 1 (n > 1) possui no mínimo 1 e no máximo n – 1 folhas.
3. Prove que o número de subárvores vazias de uma árvore binária com n nós é n + 1.
4. Considere T uma árvore binária completa com n > 0 nós. Então T possui altura h mínima. Além disso, $h=1+\lfloor \log n \rfloor$. Prove.

5. Seja T uma árvore binária completa com n nós e altura h. Então, $2^{h-1} \leq n \leq 2^h - 1$. Prove.

6. Defina: árvore estritamente m-ária; árvore m-ária completa e árvore m-ária cheia.
7. Faça análise de complexidade dos algoritmos de percurso em árvore binária (pré-ordem, ordem simétrica e pós-ordem).
8. Implemente uma versão iterativa para os algoritmos de percurso em árvore binária (pré-ordem, ordem simétrica e pós-ordem).
9. Faça um algoritmo de percurso pré-ordem para uma árvore qualquer com a representação apresentada em sala. Faça a versão recursiva e a versão iterativa.
10. Faça um algoritmo para calcular a altura de todos os nós de uma árvore binária. Considere que o nó da árvore terá um campo para armazenar a altura.
11. Qual a complexidade do algoritmo para calcular a altura de todos os nós da árvore binária?
12. Faça um algoritmo de percurso em nível de uma árvore binária.
13. Faça um algoritmo para obter o sucessor de um nó em uma árvore binária de busca.
14. Faça um algoritmo para obter o predecessor de um nó em uma árvore binária de busca.
15. Dada uma lista com n chaves, construir a árvore de busca binária de altura mínima utilizando o algoritmo de inserção. Qual a complexidade do seu método?
16. Escreva o algoritmo de remoção em árvores binárias de busca.
17. Verifique se as sequências abaixo correspondem a um heap max(min):
    a. 20, 25, 32, 29, 27, 35, 40, 45
    b. 20, 32, 25, 29, 27, 35, 40, 45
    c. 50, 45, 48, 29, 15, 35, 40, 27
18. Seja a lista: 40, 25, 72, 13, 14, 95, 48, 100, 12, 93. Determinar o heap max obtido pelo algoritmo de construção.
19. Escrever os procedimentos subir e descer, relacionados a heaps, não recursivos. Qual a complexidade destes procedimentos?
20. Descreva o funcionamento do algoritmo heapsort.

21. O professor Girafales acha que descobriu uma propriedade interessante das árvores binárias de busca. Suponha que a busca por uma chave *k* em uma árvore binária de busca termine em um nó folha. Considere três conjuntos de chaves: *A*, as chaves à esquerda do caminho de busca; *B*, as chaves no caminho de busca e *C*, as chaves à direita do caminho de busca. Professor Girafales acredita que quaisquer três chaves *a*, *b* e *c* tais que $a \in A, b \in B, c \in C$ devem satisfazer $a \leq b \leq c$. Mostre que a hipótese do professor Girafales está errada fornecendo o menor contra-exemplo possível.
22. Suponha o seguinte algoritmo para ordenar um conjunto de *n* valores:
    a. Construa uma árvore binária de busca em que os *n* valores do conjunto são as chaves dos nós da árvore (chamando o algoritmo de inserção para cada um dos nós).
    b. Faça um percurso em ordem simétrica nesta árvore, imprimindo as chaves no momento em que são visitadas.

    As chaves serão impressas em ordem crescente dos *n* valores informados. Qual é o melhor caso e o pior caso, em termos de tempo de execução, para este algoritmo de ordenação?

23. Faça um algoritmo que permite o percurso em uma árvore *m*-ária qualquer nível por nível. Ou seja, a raiz é visitada primeiro, depois os filhos da raiz (da esquerda para a direita), depois os netos da raiz (também da esquerda para a direita) e assim por diante, até as folhas. (Dica: use uma fila).
24. Implemente um método para retornar o maior valor armazenado em uma árvore de chaves naturais. Esse método deve retornar -1 caso a árvore seja vazia.
25. Faça um algoritmo para calcular o número de descendentes de cada nó em uma árvore binária.
26. Qual o menor e o maior número de elementos em uma *heap* de altura *h*?
27. Onde deve ficar o menor elemento de uma heap-max?
28. Mostre o comportamento (passo a passo) da ordenação por *heap* para a lista: 80, 35, 20, 60, 42, 36, 85.
29. O que acontece se o algoritmo *heapsort* for utilizado em uma lista já ordenada? E se a tabela estiver em ordem invertida?
30. Explique porque o *heapsort* não é um método de ordenação estável.

31. Suponha uma floresta criada pela operação gerar de conjuntos disjuntos para o conjunto {1, 2, 3, 4, 5}. Determinar a floresta obtida após a realização das seguintes operações: fundir(1,3), fundir(2,5), fundir(3,4), supondo que a operação seja realizada (i) sem critério de tamanho; (ii) com critério de tamanho.
32. Suponha uma floresta criada pela operação gerar de conjuntos disjuntos para o conjunto {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}, mostre a floresta resultante da seguinte sequência de operações de união (fusão): 8-2 9-1 0-7 6-7 7-2 3-4 4-9 4-6 3-5, considerando o critério de tamanho.
33. Dada uma árvore binária que representa uma expressão aritmética (considerando-se apenas operações binárias), gerar a mesma expressão em notação completamente parentizada.
34. Escreva um algoritmo para determinar o número de nós das subárvores de v, para cada nó v de uma árvore binária.
35. Seja a lista dada pelas prioridades a seguir: 18, 25, 41, 34, 14, 10, 52, 50, 48. Determinar o heap obtido pela aplicação do algoritmo de construção.