MAC 5710 - Estruturas de Dados e sua Manipulação

Primeiro semestre de 2001

Lista 3

1. Qual é a diferença entre a propriedade de uma árvore de busca binária e um *heap*? Dado um heap, é possível imprimir os valores nele contidos em ordem crescente em tempo $O(n)$? Explique como ou porque não.

2. Escreva um algoritmo não recursivo que recebe uma árvore binária e imprime os elementos contidos na árvore em in-ordem.

3. O Prof. Xis acha que descobriu uma propriedade muito interessante de árvores de busca binária. Suponha que a busca por um elemento x em uma árvore de busca binária termina em uma folha. Considere três conjuntos: o conjunto A com os valores da árvore à esquerda do caminho percorrido na busca, o conjunto B com os valores deste caminho, e o conjunto C, com os valores à direita deste caminho. O Prof. Xis afirma que quaisquer três valores $a \in A, b \in B$ e $c \in C$ satisfazem $a \leq b \leq c$. Dê um contra-exemplo menor possível para a afirmação do Prof. Xis.

4. Podemos ordenar um conjunto de n números primeiramente construindo uma árvore de busca binária contendo estes números e então imprimindo o conteúdo da árvore em in-ordem. Qual é o tempo de execução para este algoritmo de ordenação no pior e no melhor caso? Explique sua resposta.

5. A operação de remoção de uma árvore de busca binária, como vista em aula, é "comutativa" no sentido de que deletar x e depois y resulta na mesma árvore que deletar y e depois x? Explique porque ou dê um contra-exemplo.

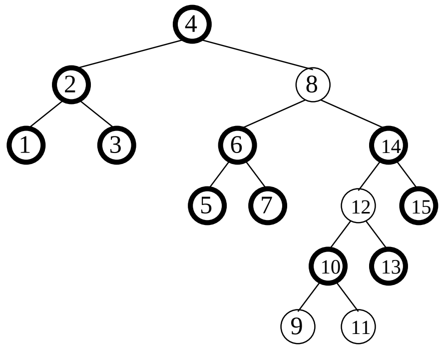
6. Mostre que a noção de árvore de busca binária aleatória com n elementos, onde cada árvore de busca binária com n elementos é igualmente provável de ser escolhida, é diferente da noção de árvore de busca binária construída aleatoriamente como visto em aula. (Dica: liste todas as possibilidades e probabilidades correspondentes para o caso $n = 3$.)

7. Escreva as funções inserção e remoção para árvores binárias com fios.

8. Escreva um algoritmo recursivo que imprima o conteúdo de todos os nós do nível i de uma árvore binária. Idem para árvores binárias com fios (usando os fios, claro). Idem para árvores arbitrárias armazenadas com a estrutura *left-child/right-sibling*, vista em aula.

9. Considere uma árvore rubro-negra inicialmente vazia. Mostre o resultado da inserção nesta árvore das chaves $1, 2, \ldots, 10$ nesta ordem. Mostre como fica a árvore após cada inserção.

10. Considere a árvore rubro-negra abaixo (os nós mais escuros são pretos e não desenhamos as folhas (nós NULL)).



Mostre a árvore rubro-negra resultante da remoção da árvore acima dos valores 1, 7, 13. Faça sempre a remoção da árvore acima, não do resultado da remoção anterior para facilitar a correção (evitando acumulo de erros...).

11. Suponha que inserimos um valor x em uma árvore rubro-negra e logo em seguida o removemos. A árvore resultante é igual à árvore rubro-negra final? E no caso de árvores AVL: se inserimos um valor x em uma árvore AVL e logo em seguida o removemos, a árvore AVL resultante é igual à inicial? Justifique as suas respostas.

12. Na figura seguinte, mostramos quatro possíveis operações executadas no procedimento de remoção de um nó de uma árvore rubro-negra. Argumente que o valor da altura rubro-negra dos nós desenhados continua o mesmo antes e depois das modificações efetuadas.