

Pesquisa e Classificação de Dados

Lista 3 (BST e AVL)

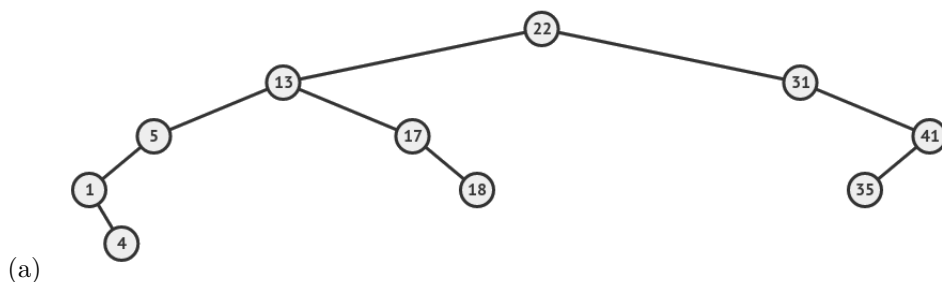
Prof. Ricardo Oliveira

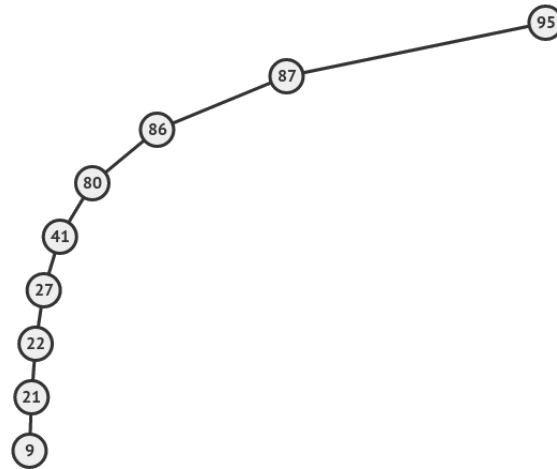
Esta lista **não** vale nota e **não** deve ser entregue, mas apenas utilizada como material de apoio para estudo. Naturalmente, você pode tirar eventuais dúvidas com o professor.

Exercícios marcados com (B) são básicos e essenciais para a matéria. Exercícios marcados com (C) são complementares. Recomenda-se fortemente a resolver todos os exercícios.

1. (B) Em uma BST inicialmente vazia, insira, nesta ordem, as chaves 50, 28, 76, 60, 12, 90, 55, 81, 8, 15, 62, 92, 57, 11 e 61 e apresente a árvore resultante.
2. (B) Na árvore obtida no exercício anterior, indique:
 - (a) sua raiz;
 - (b) todas as suas folhas;
 - (c) sua altura;
 - (d) a altura da subárvore esquerda do nodo 76;
 - (e) a altura da subárvore direita do nodo 90;
 - (f) a altura da subárvore direita do nodo 28;
 - (g) o nível do nodo 62;
 - (h) o nível do nodo 50;
 - (i) os filhos do nodo 76;
 - (j) os filhos do nodo 8;
 - (k) os filhos do nodo 57;
 - (l) o pai do nodo 15;
 - (m) o pai do nodo 76;
 - (n) o pai do nodo 50;
 - (o) o avó do nodo 61;
 - (p) o avó do nodo 12;
 - (q) o irmão do nodo 60;
 - (r) o irmão do nodo 15;
 - (s) o irmão do nodo 12;
 - (t) o tio do nodo 11;
 - (u) os sobrinhos do nodo 60;
 - (v) os sobrinhos do nodo 76;
 - (x) todos os ancestrais do nodo 57;
 - (w) todos os ancestrais do nodo 8;
 - (y) todos os descendentes do nodo 12;
 - (z) todos os descendentes do nodo 76;
 - (a1) seu percurso *preorder*;
 - (b1) seu percurso *inorder*;
 - (c1) seu percurso *posorder*.
3. (C) O menor ancestral comum (*Lowest Common Ancestor* (LCA)) de dois nodos é o nodo de maior nível (mais “baixo”) que é ancestral de ambos simultaneamente. Se ambos os nodos estão no mesmo ramo, então o LCA é o nodo de menor nível dentre os dois. Na árvore obtida no exercício anterior, determine:
 - (a) $LCA(55, 62)$
 - (b) $LCA(57, 61)$
 - (c) $LCA(11, 15)$
 - (d) $LCA(8, 90)$
 - (e) $LCA(62, 81)$
 - (f) $LCA(55, 76)$
 - (g) $LCA(15, 28)$

4. (C) Apresente um algoritmo que, dados dois nodos a e b , determina $LCA(a, b)$. Analise a complexidade de pior caso do algoritmo apresentado.
5. (C, desafio) Apresente um algoritmo de pior caso $O(\lg N)$ para obter o LCA de dois dados vértices (em qualquer BST , inclusive as desbalanceadas!) após um pré-processamento em $O(N \lg N)$.
6. (B) Apresente a BST cujo percurso *preorder* é 42, 28, 14, 1, 30, 29, 72, 51, 81.
7. (B) Na árvore apresentada na questão anterior:
 - (a) remova o nodo 1 e apresente a BST resultante;
 - (b) remova o nodo 30 e apresente a BST resultante;
 - (c) remova o nodo 72 e apresente a BST resultante;
 - (d) remova o nodo 42 e apresente a BST resultante.
8. (B) Apresente a BST cujo percurso *posorder* é 5, 9, 7, 20, 13, 12, 10.
9. (B) Apresente uma BST o *menos balanceada possível* contendo as chaves 84, 10, 58, 25, 69 e 24;
10. (C) Apresente um algoritmo que, dado um vetor de N chaves inteiras, construa uma BST o *menos balanceada possível* contendo as chaves contidas no vetor.
11. (C) Analise a complexidade de tempo de pior caso do algoritmo apresentado na questão anterior.
12. (B) Apresente uma BST o *mais balanceada possível* contendo as chaves 84, 10, 58, 25, 69 e 24;
13. (B) Apresente um algoritmo que, dado um vetor de N chaves inteiras, construa uma BST o *mais balanceada possível* contendo as chaves contidas no vetor.
14. (B) Analise a complexidade de tempo de pior caso do algoritmo apresentado na questão anterior.
15. (C) Sejam A e B duas $BSTs$ tais que todas as chaves de B são maiores que todas as chaves de A . Apresente um algoritmo que una as duas árvores, isto é, resulte em uma BST contendo todas as chaves de ambas as árvores. Apresente o algoritmo mais eficiente que puder.
16. (C) Analise a complexidade de tempo de pior caso para o algoritmo anterior.
17. (B) Apresente o balanceamento de cada nodo em cada árvore a seguir:





- (b)
18. (C) Mostre que toda árvore AVL com N nodos tem altura $h = \Theta(\lg N)$.
 19. (C) Prove que, durante a manutenção de uma árvore AVL, nenhum nodo chegará a ter balanceamento ± 3 .
 20. (C) Seja z um nodo desbalanceado (isto é, com balanceamento 2 ou -2) após a inserção de algum nodo novo em uma árvore AVL. Mostre que:
 - (a) Se a inserção foi feita em sua subárvore esquerda, então z tem balanceamento 2 (e não -2);
 - (b) Se a inserção foi feita em sua subárvore direita, então z tem balanceamento -2 (e não 2).
 21. (B) Em uma árvore AVL inicialmente vazia, insira, nesta ordem, os elementos 50, 60, 70, 40, 30, 65, 67, 55 e 52. Após cada inserção, indique as rotações realizadas e apresente a árvore AVL resultante.
 22. (B) Em uma árvore AVL inicialmente vazia, insira, nesta ordem, os elementos 1, 2, 3, 4, 5, 6, 7, 8 e 9. Após cada inserção, indique as rotações realizadas e apresente a árvore AVL resultante.
 23. (B) Em uma árvore AVL inicialmente vazia, insira, nesta ordem, os elementos 8, 4, 10, 1, 5, 9 e 12. Após cada inserção, indique as rotações realizadas e apresente a árvore AVL resultante.
 24. (B) Em uma árvore AVL inicialmente vazia, insira, nesta ordem, os elementos 65, 3, 58, 25, 36, 80, 16, 85, 50, 8, 18, 39, 48, 56, 92, 23, 15 e 38. Após cada inserção, indique as rotações realizadas e apresente a árvore AVL resultante.
 25. (C) Apresente um algoritmo que, dados uma árvore AVL com N chaves e um inteiro k , determina a k -ésima menor chave da árvore.
 26. (C) Analise a complexidade de tempo de pior caso do algoritmo obtido no exercício anterior. Caso não seja $O(\lg N)$, apresente outro algoritmo que o seja.
 27. (C) Considere seu algoritmo indicado no exercício 15. Se A e B forem ambas AVLs, a árvore resultante será uma AVL? Justifique.
 28. (C) Considere seu algoritmo indicado no exercício 15. Se A e B forem ambas AVLs e *tiverem a mesma altura*, a árvore resultante será uma AVL? Justifique.