

Logo, o espaço total para descrever a função *hpm* gerada pelo Programa 5.42 é $(2 + \epsilon)cn$ bits. Usando $c = 1,23$ e $\epsilon = 0,125$, a função *hash* perfeita mínima necessita aproximadamente 2,62 bits por chave para ser armazenada.

Assim, uma função *hash* perfeita mínima pode ser armazenada em aproximadamente 2,62 bits por chave. Mehlhorn (1984) mostrou que o limite inferior para armazenar uma função *hash* perfeita mínima é $N \log e + O(\log N) \approx 1,44N$. Assim, o valor de aproximadamente 2,62 bits por chave é um valor muito próximo do limite inferior de aproximadamente 1,44 bits por chave para essa classe de problemas.

O principal resultado desta seção mostra um algoritmo prático que reduziu a complexidade de espaço para armazenar uma função *hash* perfeita mínima de $O(N \log N)$ bits para $O(N)$ bits. Isso permite o uso de *hashing* perfeito em aplicações em que antes não eram consideradas uma boa opção. Por exemplo, Botelho, Lacerda, Menezes e Ziviani (2009) mostraram que uma função *hash* perfeita mínima apresenta o melhor compromisso entre espaço ocupado e tempo de busca quando comparada com todos os outros métodos de *hashing* para indexar a memória interna para conjuntos estáticos de chaves.

Notas Bibliográficas

As principais referências para pesquisa em memória interna são Gonnet e Baeza-Yates (1991), Knuth (1973) e Mehlhorn (1984). Outros livros incluem Standish (1980), Wirth (1976; 1986), Aho, Hopcroft e Ullman (1983), Terada (1991). Um estudo mais avançado sobre estruturas de dados e algoritmos pode ser encontrado em Tarjan (1983).

Um dos primeiros estudos sobre inserção e retirada em árvores de pesquisa foi realizado por Hibbard (1962), tendo provado que o comprimento médio do **caminho interno** após n inserções randômicas é $2 \ln n$. A definição de árvore binária foi extraída de Knuth (1968, p. 315).

A primeira árvore binária de pesquisa com balanceamento foi proposta por Adel'son-Vel'skii e Landis (1962), dois matemáticos russos, a qual recebeu o nome de árvore AVL. Uma árvore binária de pesquisa é uma **árvore AVL** se a altura da subárvore à esquerda de cada nó nunca difere de ± 1 da altura da subárvore à direita. A Figura 5.20 apresenta uma árvore com esta propriedade.

A forma de manter a propriedade AVL é por meio de transformações localizadas no caminho de pesquisa. Como a altura das árvores AVL fica sempre entre $\log_2(n+1)$ e $1.4404 \log_2(n+2) - 0.328$ (Adel'son-Vel'skii e Landis, 1962), o custo para inserir ou retirar é $O(\log n)$, que é exatamente o custo para percorrer o caminho de pesquisa. Wirth (1976; 1986) apresenta implementações dos algoritmos de inserção e de retirada para as árvores AVL.

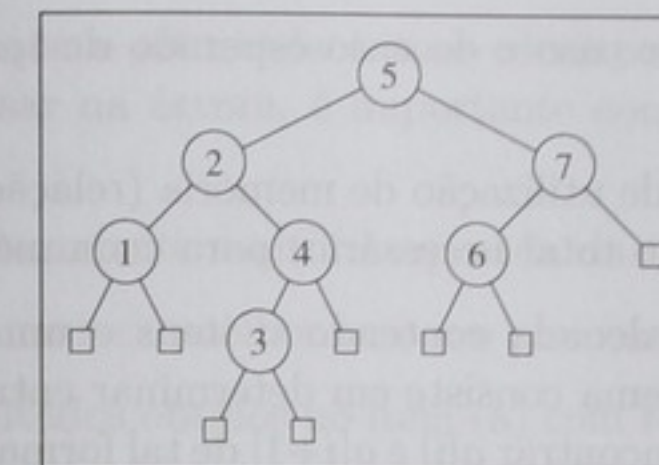


Figura 5.20 Árvore AVL.

O material utilizado na Seção 5.3.2 veio de Bayer (1971; 1972), Olivé (1980), Ziviani e Tompa (1982) e Ziviani, Olivé e Gonnet (1985). Os trabalhos de Bayer apresentam as árvores SBB, o de Olivé sugere uma melhoria para o algoritmo de inserção, e o de Ziviani e Tompa apresentam implementações para os algoritmos de inserção e retirada. A árvore SBB pode ser vista como uma representação binária da **árvore 2-3-4**, apresentada por Guibas e Sedgewick (1978). Este mesmo trabalho mostra como adaptar vários algoritmos clássicos para árvores de pesquisa balanceadas dentro do esquema **árvores red-black**.

Sleator e Tarjan (1983) apresentam vários métodos para manutenção de **árvores autoajustáveis**. A ideia é mover os nós mais frequentemente acessados em direção à raiz após cada acesso: embora cada operação isolada possa ter custo mais alto, ao longo de um período maior, o tempo médio de cada operação é menor, isto é, o **custo amortizado** diminui ao longo do tempo. Em outras palavras, uma operação particular pode ser lenta, mas qualquer sequência de operações é rápida.

A principal referência sobre *hashing* é Knuth (1973). Existem várias propostas para a construção de funções de transformação perfeitas, como em Fox, Heath, Chen e Daoud (1992). As principais referências utilizadas na seção sobre *hashing* perfeito com ordem preservada são Czech, Havas e Majewski (1992, 1997). Outras referências são Majewski, Wormald, Havas e Czech (1996) e Witten, Moffat e Bell (1999). As principais referências utilizadas na seção sobre *hashing* perfeito com espaço quase ótimo são Botelho (2008), Botelho e Ziviani (2007), Botelho, Pagh e Ziviani (2007) e Botelho, Lacerda, Menezes e Ziviani (2010). O algoritmo proposto por Botelho (2008) é o primeiro algoritmo prático descrito na literatura que utiliza $O(1)$ bits por chave para uma função *hash* perfeita mínima.

Exercícios

1. Considere as técnicas de pesquisa sequencial, pesquisa binária e a pesquisa baseada em *hashing*.
 - a) Descreva as vantagens e desvantagens de cada uma dessas técnicas, indicando em que situações você usaria cada uma delas.