

MAC 5710 – Estrutura de Dados e sua Manipulação

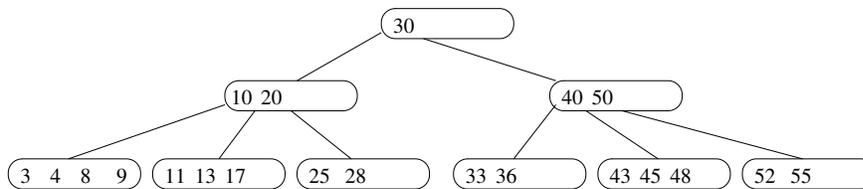
Primeiro semestre de 2007

Lista de exercícios 4 — Data máxima para entrega: 18/06/2007

OBS.: Entregar apenas os exercícios pontuados.

1. Insira a seqüência de chaves 39, 48, 61, 85, 17, 21, 8, 25, 42, 70, 77, 31, 99, 96, 88, 74, 66, 3, 15, 21 em cada uma das árvores a seguir, inicialmente vazias:
 - a) árvore binária de busca
 - b) AVL
 - c) árvore rubro-negra
 - d) árvore 2-3
 - e) B-árvore, com $t=2$
 - f) B-árvore, com $t=3$
2. Na três primeiras árvores geradas no exercício anterior, calcule
 - a) o comprimento de caminho interno
 - b) o comprimento de caminho externo.
3. Sejam $I(T)$ e $E(T)$ os comprimentos de caminho interno e externo, respectivamente, de uma árvore binária de busca T . Mostre que se n é o número de nós de T , então $E(T) = I(T) + n$.
4. Explique por que as operações de rotação preservam a in-ordem em árvores binárias.
5. (2 pontos) Mostre que uma árvore binária qualquer com n nós pode ser transformada em qualquer outra árvore binária com n nós usando $O(n)$ operações de rotação. (Dica: Mostre primeiramente que no máximo $n - 1$ rotações são suficientes para se transformar qualquer árvore binária em uma árvore binária linear, com todos os filhos esquerdo igual a NIL).
6. Dê exemplo de uma família de árvores AVL cuja remoção de nós implica a realização de $O(\log n)$ operações de rotação para o rebalanceamento.
7. Desenhar todas as árvores RB com 5 nós internos.
8. Mostre que o caminho mais longo de um nó x a um nó folha em uma árvore RB é no máximo duas vezes o comprimento do caminho mais curto entre x e um nó folha.
9. (1 ponto) Prove ou dê um contra-exemplo para a seguinte afirmação: “Seja T uma árvore RB cujo nó raiz tem a cor rubra. Se esta for alterada para negra, a árvore mantém-se rubro-negra.”
10. No algoritmo de inserção de nó numa árvore RB visto em sala de aula (Cormen), o nó raiz sempre recebe a cor BLACK. Qual a vantagem de se fazer isso?
11. (2 pontos) Suponha que um elemento x é inserido numa árvore RB e é removida imediatamente em seguida (usando os algoritmos descritos em Cormen). A árvore resultante após a remoção é exatamente igual à árvore antes da inserção? Explique.

12. Desenhe uma B-árvore de grau ¹ $t = 3$ obtida inserindo-se as seguintes chaves: 1, 3, 6, 8, 14, 32, 36, 38, 39, 41, 43. Se a ordem de inserção for alterada, a árvore resultante é diferente? Mostre um exemplo em caso afirmativo, ou prove em caso negativo.
13. (2 pontos) Prove ou dê um contra-exemplo para a seguinte afirmação: “Para qualquer conjunto de chaves e qualquer inteiro $t > 1$, sempre existe uma B-árvore de grau t que armazena essas chaves.”
14. Qual é o número mínimo e máximo de nós que uma B-tree de grau t pode ter, supondo que ela armazena m chaves?
15. Mostre como fica a B-árvore abaixo ($t=3$) ao serem removidos os elementos 45, 30, 28, 50, 8, 10, 4, 20, 40, 55, 17, 33, 11, 36, 3, 11, 52 (nessa ordem), de acordo com o algoritmo descrito em Cormem e de acordo com o outro algoritmo.



Mostre algumas configurações intermediárias e não apenas a árvore final.

16. Verifique se a seqüência 33 32 28 31 26 29 25 30 27 corresponde a um heap.
17. (2 pontos) Seja um heap T e sua respectiva representação linear no vetor A . Sejam $A[i]$ e $A[j]$ dois elementos desse heap tais que $i < j$ e $A[i] < A[j]$. Se trocarmos $A[i]$ e $A[j]$ entre si, a nova seqüência linear em A é um heap? E se em vez de $A[i] < A[j]$, tivéssemos $A[i] > A[j]$?
18. Num heap, além das operações de inserção, máximo e remoção (de máximo), podemos considerar também a operação alteração de prioridade. A prioridade de um nó pode ser aumentada ou pode ser diminuída. Dado o índice i de um elemento do heap, explique como pode ser realizado o aumento de sua prioridade (isto implica em mudar alguns elementos de posição para que a propriedade heap continue válida). Escreva um algoritmo que realiza a operação de aumento de prioridade de um nó com índice i .
19. Repita o exercício anterior para a diminuição de prioridade.
20. Dado o heap representado pela seqüência 92 85 90 47 91 34 20 4 46, qual o heap resultante se alteramos a prioridade do quinto nó de 91 para 93?
21. Seja T uma árvore binária completa (no sentido da árvore heap vista em aula) e seja A a sua representação linear em vetor (ou seja, $A[1]$ corresponde ao nó raiz, $A[2]$ ao filho esquerdo do nó raiz, $A[3]$ ao filho direito do nó raiz, etc). Então dizemos que T é um *heap min-max* se
 - i) a chave armazenada no nó raiz (ou seja, $A[1]$) é a menor de todas
 - ii) para todo índice $i > 1$ correspondendo aos nós em níveis ímpares (nível mínimo) da árvore,

$$A[i] \leq A[\lfloor i/2 \rfloor]$$

¹Grau de uma B-árvore é o número mínimo de filhos permitidos para os nós internos, exceto a raiz. Se o grau de uma B-árvore é t , então o número mínimo de entradas permitidas por nó é $t - 1$.

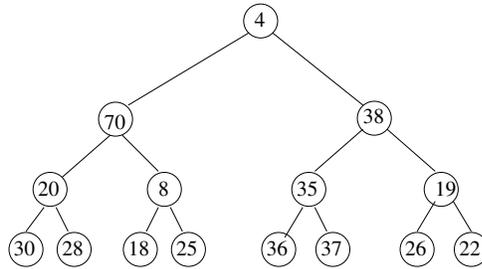
$$A[i] \geq A[\lfloor i/4 \rfloor], i \geq 4$$

iii) para todo índice $i > 1$ correspondendo aos nós em níveis pares (nível máximo) da árvore,

$$A[i] \geq A[\lfloor i/2 \rfloor]$$

$$A[i] \leq A[\lfloor i/4 \rfloor], i \geq 4$$

Exemplo:



- Mostre que a prioridade do nó correspondente ao índice i , localizado num nível ímpar de um heap min-max, é menor que a de seus descendentes.
- Escreva um algoritmo para inserção de um elemento num heap min-max
- Escreva um algoritmo para remoção do elemento mínimo de um heap min-max
- Escreva um algoritmo para remoção do elemento máximo de um heap min-max