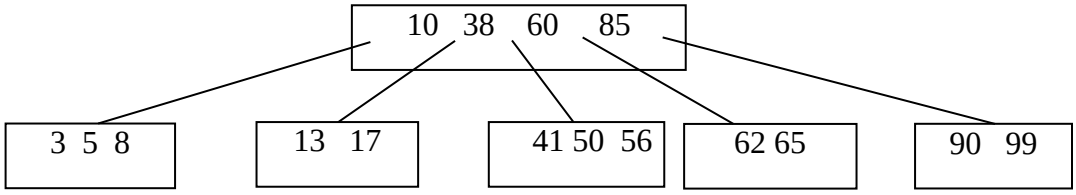
Universidade Federal do Rio de Janeiro
Instituto de Matemática - DCC
Organização de Dados I **13/06/2012**

Prova 2

1.1 (1.0) – Insira as seguintes chaves na arvore B de ordem 2 abaixo: 39, 77, 42, 73. Mostre a arvore antes e após **cada** insercao que altere a estrutura da arvore.



1.2 (1.0) – Na árvore inicial desenhada acima, remova as chaves 3, 10 e 90.

2.1 (1.5)– Escreva um a função DESCER, semelhante a da heap binaria, para uma heap binomial.

2.2 (0.5)– Explique qual a complexidade deste algoritmo.

3 (1.0) - Construa uma arvore de huffman para a seguinte string: “o carro correu na corrida”. Qual o comprimento da string, quando codificada utilizando esta arvore.

4.1 (1.0)- Insira as seguintes chaves: 1, 12, 6 , 7, 11, 2, 9 e 18 em uma tabela hash utilizando a funcao de hash h(x) = (3x + 1) mod 10 com enderecamento interno sem area de colisao, e com coalescencia.

4.2 (1.0)- Repita as insercoes do item anterior, agora em uma tabela sem coalescencia.

5- Considere 10 conjuntos inicialmente unitarios, com elementos 1, 2, 3, ... 10, com **sem uniao por tamanho, e COM compressao de caminhos. Uma sempre o primeiro conjunto ao segundo.**

5.1 (1.0)- Mostre como ficam os conjuntos apos as operacoes: une(1,2), une(2,3), une(3,4), une(4,5), une(6,7), une(7,8), une(8,9), une(9,10), une (2,7). Mostre como fica o vetor que representa o conjunto ao final das operacoes. LEMBRE-SE da compressao de caminhos no find !

5.2 (1.0)- Escreva o algorimo de uniao( ) com criterio de tamanho, e compare a execucao deste com o mesmo algoritmo sem este criterio, para as chaves acima.

6 (1.0). Insira as seguintes chaves em uma árvore digital binaria: 01001, 11001, 1111, 0100, e 101010.