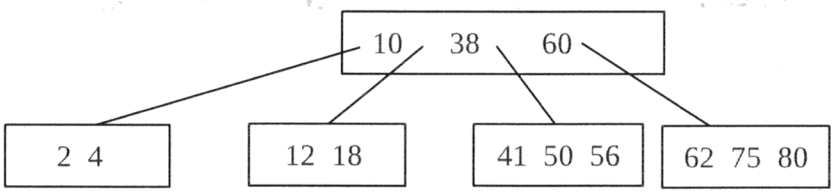
Universidade Federal do Rio de Janeiro
Instituto de Matemática - DCC
Organização de Dados I **04/12/2013**

Prova 2

1.1 (1.0) – Insira as seguintes chaves na arvore B de ordem 2 abaixo: 82, 77, 45, 7 e 42. Mostre a arvore antes e após **cada** insercao que altere a estrutura da arvore.



2.1 (1.0)– Escreva um a função que dado um nó de uma heap de Fibonacci, encontre o seu nó filho de menor chave.

2.2 (0.5) – Utilizando o algoritmo do item anterior, implemente uma operação de DESCER na árvore, semelhante a da heap binária.

2.3 (0.5)– Explique qual a complexidade deste algoritmo.

3.1 (1.0) Insira as seguintes chaves em ordem em uma heap de Fibonacci SEM avaliacao tardia: 10, 1, 13, 15, 17, 22, 5 , 16 e 43.

3.2 (1.0) Diminua a prioridade da chave 16 para 2 e 22 para 2.

4.1 (1.0)- Insira as seguintes chaves: 1, 11, 4 , 13, 12, 2, 9 e 5 em uma tabela hash utilizando a funcao de hash h(x) = x mod 10 com enderecamento interno sem area de colisao, e sem coalescencia.

4.2 (1.0)- Repita as insercoes do item anterior, agora em uma tabela com enderecamento aberto, e funcao de dispersão dada por h(x,k) = (x + k) mod 10.

5 (1.0) - Construa uma arvore de huffman para a seguinte string: "vovo viu a uva". Quantos bits seriam utilizados para representar esta string utilizando a arvore obtida ?

6- Considere 10 conjuntos inicialmente unitarios, com elementos 1, 2, 3, ... 10, com **uniao por rank, e com compressao de caminhos. Uma sempre o primeiro conjunto ao segundo em caso de empate nos criterios.**

6.1 (1.0)- Mostre como ficam os conjuntos apos as operacoes: une(1,2), une(2,3), une(4,5), une(6,7), une(4,7), une(1,4), une(9,10). Mostre como fica o vetor que representa o conjunto ao final das operacoes. LEMBRE-SE da compressao de caminhos no find !

6.2 (1.0) – Considere operações de uniao que são feitas recebendo apenas os nós que são raizes de seus conjuntos. Suponha portanto, que a união não chama o procedimento de find. Mostre que a complexidade amortizada de uma sequencia de unioes e finds, tal que toda união ocorre antes dos finds é constante. Descreva claramente a função potencial utilizada.