UNIVERSIDADE DE SÃO PAULO
INSTITUTO DE CIÊNCIAS MATEMÁTICAS E DE COMPUTAÇÃO
I.C.M.C.

SCC-601 Algoritmos e Estruturas de Dados I (EC)
Profa. Graça Nunes
2º. Semestre de 2010

3ª. Prova (Gabarito)
02/12/2010

Nome: _____ Nro USP: _____

1.(1.0) Por que nos damos ao trabalho de procurar trabalhar com árvores binárias balanceadas? Justifique.

Pois as árvores balanceadas são aquelas que têm altura mínima (se n nós, altura = $\log_2 n$) e, em tarefas como a busca por uma chave numa árvore de busca binária, isso significa que o número máximo de comparações corresponde à altura da árvore, portanto, $\log_2 n$ – que é menor do que n que é o custo de uma busca sequencial.

2. (2.0) Considere:

```
typedef struct no{
   tipo_elem info;
   struct no* esq;
   struct no* dir;
}no;
typedef struct no *pno;
```

O que faz a função abaixo?

```
pno surpresa(pno t){
(* retorna uma árvore que é a imagem no espelho de t *)
if (Vazia(t)) return Null;
else {
    pno p = (pno) malloc(sizeof (no));
    p->info = (t)->info;
    p->esq = surpresa(t->dir);
    p->dir = surpresa(t->esq);
    return p;
 }
```

3. (2.0) As afirmações abaixo são verdadeiras ou falsas? Justifique.

(i) São necessárias menos operações para inserir os elementos de um vetor ordenado em uma AVL (percorrendo-se o vetor da esquerda para a direita) do que se o vetor não estiver ordenado.

FALSO. Por ser uma sequência ordenada, mais operações de rebalanceamento serão necessárias.

(ii) Para qualquer conjunto de dados, AVL é sempre mais eficiente do que uma ABB.

FALSO. Se a ABB estiver perfeitamente balanceada, a busca pode ser até mais rápida do que na AVL, já que sua altura vai ser a mínima possível.

(iii) Busca binária em array ordenado e busca em uma ABB são igualmente eficientes no pior caso.

FALSO. Seria verdade se fosse AVL e não ABB. Numa ABB podemos ter o pior caso igual a O(n), quando ela for degenerada.

(iv) AVLs são ABBs perfeitamente balanceadas. Se fossem apenas balanceadas, teriam complexidade linear na busca de chaves.

FALSO. AVLs são ABBs balanceadas e têm complexidade log n

4. Para a árvore binária abaixo:

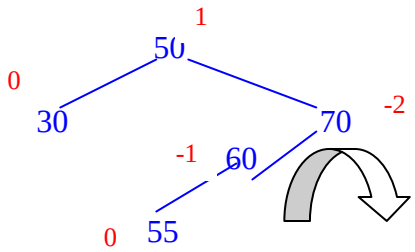
(1 (2 (4) (5)) (3 (6) (7)))

(a) (0.5) Diga se é balanceada, perfeitamente balanceada ou nenhum dos casos ou ambos;
perfeitamente balanceada e balanceada;

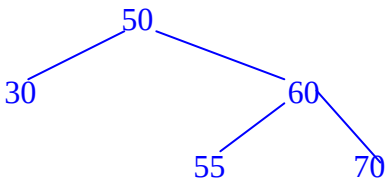
(b) Liste seus nós em
(i) (0.5) pré-ordem
1 2 4 5 3 6 7
(ii) (0.5) in-ordem
4 2 5 1 6 3 7
(iii) (0.5) pós-ordem
4 5 2 6 7 3 1

5. Desenhe a árvore AVL resultante das operações (é acumulativo: o item (**i**) ocorre na AVL vazia; os demais, nas árvores resultantes dos itens imediatamente anteriores), considerando as operações de rebalanceamento dadas em aula, no caso das inserções. Indique quando e quais rotações foram necessárias.

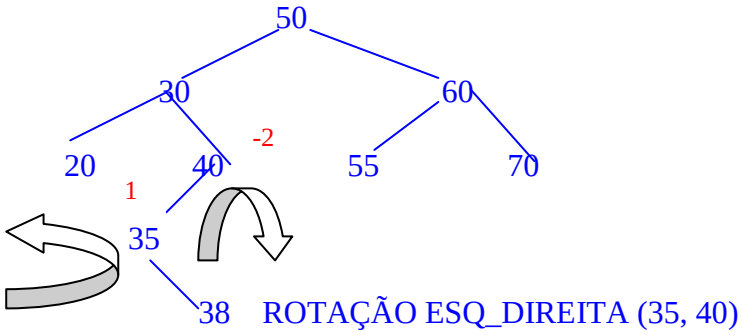
(i) (1.0) Inserção das chaves: 50, 70, 30, 60, 55;

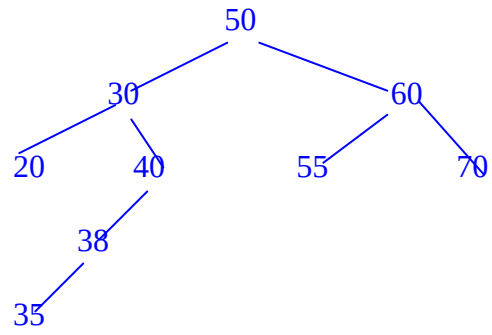


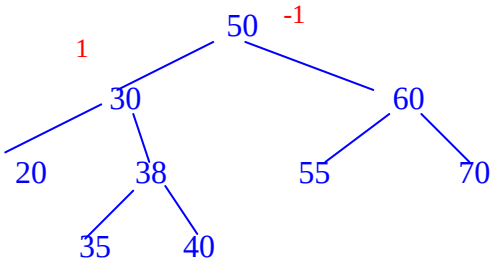
ROTAÇÃO À DIREITA 70



(ii) (1.0) Inserção das chaves: 40, 20, 35, 38







(iii) (1.0) Eliminação das chaves: 40, 50, 38: Substitui pelo menor dos maiores (55) ou pelo maior dos menores (40)