

## Estrutura de Dados para Automação Exercícios sobre Árvores AVL

## Prof. Rodrigo da Silva Guerra

## 13 de junho de 2018

I Considerando o algoritmo abaixo que implementa a inclusão de elementos numa árvore binária de busca, responda:

```
int data[15];
                                                                 return 2*node+2;
                                                        19
                                                            }
                                                        20
2
   void init_tree()
3
                                                        21
                                                            void save_number(int root, int number)
                                                        22
                                                        23
        for (i = 0 ; i < 15 ; i++)
                                                                 if (data[root] = -1)
                                                        24
                                                        25
            data[i] = -1;
                                                                     data[root] = number;
                                                                  else {
9
                                                        27
   }
                                                                     if (number > data[root])
10
11
                                                        29
   int left_child(int node)
                                                                         save_number(right_child(root), number);
12
                                                        30
13
        return 2*node+1;
                                                                         save_number(left_child(root), number);
14
                                                        32
15
                                                                }
16
                                                        ^{34}
   int right_child(int node)
                                                            }
17
                                                        35
```

**I.a.** Observe a árvore ilustrada na Figura 1 e preencha a Tabela 1 supondo que a árvore foi construída seguindo o algoritmo de inserção apresentado (2 pontos).



Tabela 1: Array referente ao exercício I.a.

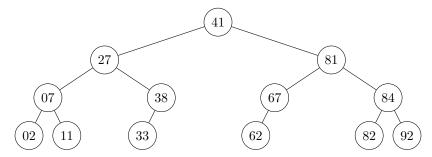
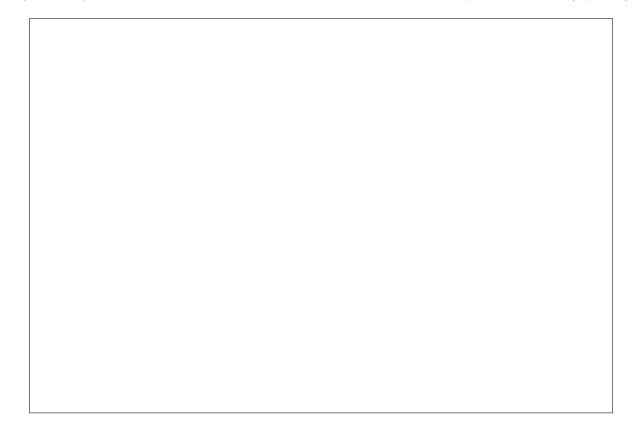


Figura 1: Árvore AVL

**I.b** O código mostrado suporta árvores de até 4 níveis. Se desejarmos alterar este código para suportar uma árvore de até 17 níveis, qual é o tamanho do array que deveria ser alocado neste caso? (2 pontos)



II.a Suponha agora que a árvore da Figura 1 é uma árvore AVL, onde o fator de equilíbrio é sempre corrigido através de rotações. Desejamos incluir o número 30. Desenhe abaixo como ficará a árvore após essa inclusão (2 pontos).



l para a fun show(int r (data[node how(right. printf("%d,	ção show(0)	(com argument);		a abaixo. Supesperada na tel:	mos uma ch
para a fun show(int r (data[node how(right. printf("%d,	ção show(0) node) e] != -1) child (node) ", data [node]	(com argument);			mos uma ch
l para a fun show(int r (data[node how(right. printf("%d,	ção show(0) node) e] != -1) child (node) ", data [node]	(com argument);			mos uma ch
l para a fun show(int r (data[node how(right. printf("%d,	ção show(0) node) e] != -1) child (node) ", data [node]	(com argument);			mos uma ch
l para a fun show(int r (data[node how(right. printf("%d,	ção show(0) node) e] != -1) child (node) ", data [node]	(com argument);			mos uma c