

1) Considere a estrutura de dados árvore binária de busca e que cada um de seus elementos possua três campos.

- left: um ponteiro para o elemento à sua esquerda;
- value: informação armazenada pelo elemento; e
- right: um ponteiro para o elemento à sua direita.

Considere ainda uma árvore binária de busca preexistente armazenada na variável *root* “e uma função *newNode* que cria um novo elemento de árvore com o valor dentro (seus ponteiros são inicializados como nulos)”.

```
func doTask(root, key)
{
    if (root == NULL)
    {
        return newNode(key);
    }

    if (key < root->value)
    {
        root->left = doTask(root->left, key);
    }
    else
    {
        root->right = doTask(root->right, key);
    }
    return root;
}
```

Analisando-se o trecho de pseudocódigo apresentado, qual é a tarefa realizada por ele?

- A) Inserir *key* no lugar do primeiro elemento da árvore binária de busca armazenada em *root*.
- B) Remover *root* da árvore binária de busca armazenada em *key*.
- C) Reordenar a árvore binária de busca armazenada em *root*.
- D) Inserir *key* na árvore binária de busca armazenada em *root*.
- E) Buscar *key* na árvore binária de busca armazenada em *root*.

2) Considere o esquema de uma Árvore Binária de BUSCA (ABB). Se inserirmos os nós 50, 20, 15, 40, 5, 62, 55, 110, 80, 72, 75 e 76

os nós folha da ABB resultante são

- A) 5, 15, 40, 55, 110, 80, 72, 75 e 76.
- B) 72, 75 e 76.
- C) 5, 40, 55 e 76.
- D) 5 e 76.
- E) 50, 20 e 62.

3) Considerando que a raiz fica no nível 1 da árvore binária de busca e que a seguinte sequência de nós foi inserida: 60, 10, 20, 80, 15, 75, 12 e 77, respectivamente. A altura da árvore e o nível do nó 15, respectivamente, serão:

- A) 5 e 5
- B) 5 e 4
- C) 6 e 4
- D) 6 e 3
- E) 7 e 4