

Avaliação: **Avaliação 2** Data: / / Curso:

Disciplina: **Estrutura de Dados** Professor: **Adriano Rivolli**

Nome: R.A.: Nota

1) (1.0) Considerando os métodos de ordenação bubble-sort e selection-sort, indique quantas trocas deverão ser realizadas para que o vetor esteja devidamente ordenado. Considere que o método de seleção não realiza trocas se o número selecionado encontra-se na posição correta.

0	1	2	3	4	5	6	7
14	15	6	7	11	10	17	2

Bubble sort: 16

Selection sort: 6

2) (2.0) Escreva uma função de ordenação (método de inserção) que modifique apenas um segmento do vetor recebido por parâmetro. Por exemplo, considere que deseja-se ordenar o vetor acima entre as posições 1 e 4, o resultado após a chamada do método é a sequência: 14, **6, 7, 11, 15**, 10, 17, 2

```
void insertion(int vet[], int ini, int fim) {
    int i, j, aux;

    for (i=ini; i<=fim; i++) {
        aux = vet[i];
        for (j=i-1; j>=ini && aux < vet[j]; j--) {
            vet[j+1] = vet[j];
        }
        vet[j+1] = aux;
    }
}
```

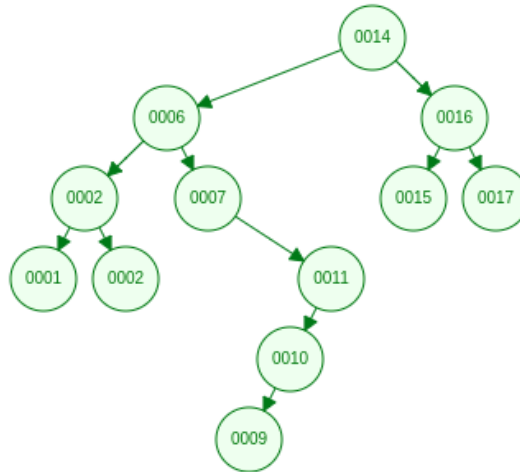
3) (2.0) Implemente uma função de busca binária que pesquise pelo código dos produtos, conforme a estrutura de dados a seguir. Assuma que o vetor encontra-se devidamente ordenado. Retorne NULL se nenhum valor for encontrado.

```
typedef struct {
    int codigo;
    char nome[];
    float preco;
} produto;

produto *buscaBinaria(produto *vetor[], int cod, int left, int right) {
    if (left > right) {
        return NULL;
    }
    int mid = (left+right) / 2;
    if (vetor[mid]->codigo == cod) {
        return vetor[mid];
    } else if (cod < vetor[mid]->codigo) {
        return buscaBinaria(vetor, cod, left, mid-1);
    } else {
        return buscaBinaria(vetor, cod, mid+1, right);
    }
}
```

4) (2.0) Considerando os conceitos de uma árvore binária de busca, faça.

- a) Desenhe a árvore considerando que a seguinte sequência de valores seja fornecida:
14, 16, 6, 7, 11, 15, 10, 17, 2, 1, 2, 9



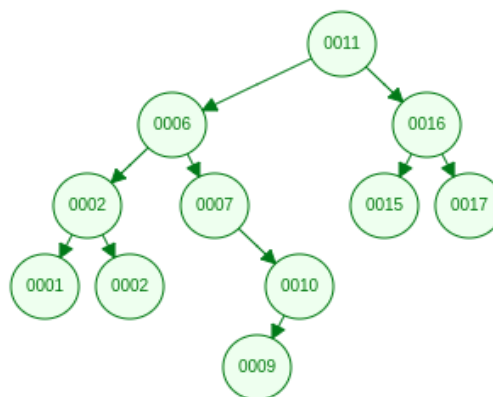
- b) Exiba o resultado do percurso de pós-ordem

1, 2, 2, 9, 10, 11, 7, 6, 15, 17, 16, 14

- c) Quais os nós folhas da árvore encontram-se no nível 3?

1 e 2

- d) Desenhe a nova árvore considerando que o nó 14 seja removido



5) (1.5) Implemente uma função para somar os valores presente em uma árvore.

```
int soma(node *tree) {  
    if (tree == NULL) return 0;  
    return info(tree) + soma(left(tree)) + soma(right(tree));  
}
```

6) (1.5) Implemente uma função para contar o número de folhas em uma árvore.

```
int contafolhas(node *tree) {  
    if (tree == NULL) return 0;  
    if (isLeaf(tree)) return 1;  
    return contafolhas(left(tree)) + contafolhas(right(tree));  
}
```

Boa prova!!!