



INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO
MARANHÃO

Sistemas de Informação

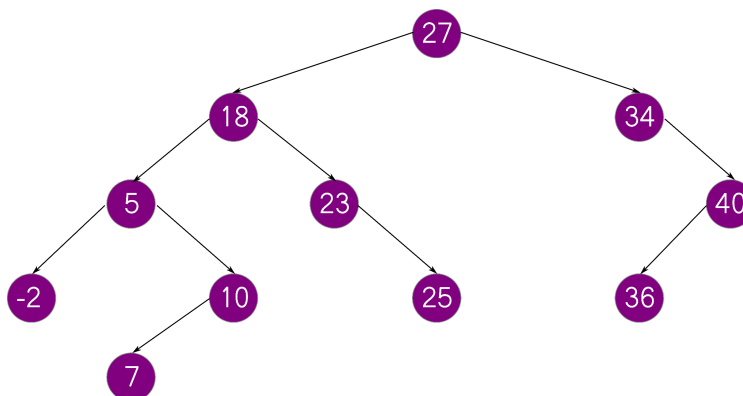
Algoritmos e Estruturas de Dados – Prof. Gentil Cutrim Serra Jr

3ª Avaliação – 13/12/2022

Aluno: _____

1) [2pt] [ENADE - 2021 - Adaptado] O uso da estrutura de dados tipo Árvore Binária de Busca é uma técnica fundamental de programação. Uma árvore binária é um conjunto finito de elementos que está vazio ou é particionado em três subconjuntos, a saber: 1) raiz da árvore - elemento inicial (único), 2) subárvore da esquerda - se vista isoladamente compõe outra árvore e 3) subárvore da direita - se vista isoladamente compõe outra árvore. A árvore pode não ter qualquer elemento (árvore vazia). A definição de árvore é recursiva e, devido a isso, muitas operações sobre árvores binárias utilizam recursão. Sendo “A” a raiz de uma árvore binária e “B” a raiz de sua subárvore esquerda ou direita, é dito que “A” é pai de “B” e que “B” é filho de “A”. Um elemento sem filhos é chamado de folha. A altura da árvore é o número de elementos encontrados no caminho descendente mais longo que liga a sua raiz até uma folha. Uma Árvore de Busca Binária é uma árvore binária especializada, na qual a informação que o elemento filho esquerdo possui é numericamente menor que a informação do elemento pai. De forma análoga, a informação que o elemento filho direito possui é numericamente maior ou igual à informação do elemento pai. O objetivo de organizar dados em Árvores Binárias de Busca é facilitar a tarefa de encontrar um determinado elemento. O percurso completo de uma árvore binária consiste em visitar todos os elementos desta árvore, segundo algum critério, a fim de processá-los. Três formas são bem conhecidas para a realização deste percurso: 1) pré-ordem, 2) em-ordem e 3) pós-ordem. Considerando o texto apresentado e que a seguinte lista de elementos numéricos: (27, 34, 40, 18, 23, 5, 25, 36, 10, 7, -2) seja totalmente transferida para uma estrutura de Árvore Binária de Busca, inicialmente vazia, elemento a elemento, da esquerda para a direita:

- monte essa árvore binária de busca.
- exiba os resultados de um percurso dessa árvore em pré-ordem.



Pré-ordem: 27, 18, 5, -2, 10, 7, 23, 25, 34, 40, 36



INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO
MARANHÃO

Sistemas de Informação

Algoritmos e Estruturas de Dados – Prof. Gentil Cutrim Serra Jr

3ª Avaliação – 13/12/2022

2) [1pt] [ENADE - 2011] Suponha que se queira pesquisar a chave 287 em uma árvore binária de pesquisa com chaves entre 1 e 1000. Durante uma pesquisa como essa, uma sequência de chaves é examinada. Cada sequência abaixo é uma suposta sequência de chaves examinadas em uma busca da chave 287.

- I. 7, 342, 199, 201, 310, 258, 287
- II. 110, 132, 133, 156, 289, 288, 287
- III. 252, 266, 271, 294, 295, 289, 287
- IV. 715, 112, 530, 249, 406, 234, 287

É válido apenas o que se apresenta em

- (A) I.
- (B) III.
- (C) I e II.
- (D) II e IV.
- (E) III e IV.

3) [1pt] [ENADE - 2014] Qual o valor de retorno da função a seguir, caso $n = 27$?

```
int recursao (int n) {  
    if (n <= 10) {  
        return n * 2;  
    }  
    else {  
        return recursao(recursao(n/3));  
    }  
}
```

Resposta:

Para $n = 27$:

$\text{recursao}(27) = \text{recursao}(\text{recursao}(27/3)) = \text{recursao}(\text{recursao}(9))$

Como $9 \leq 10$, então $\text{recursao}(9) = 2 * 9 = 18$:

$\text{recursao}(27) = \text{recursao}(18) = \text{recursao}(\text{recursao}(6))$

Como $6 \leq 10$, então $\text{recursao}(6) = 2 * 6 = 12$

$\text{recursao}(27) = \text{recursao}(12) = \text{recursao}(\text{recursao}(4))$

Como $4 \leq 10$, então $\text{recursao}(4) = 2 * 4 = 8$



INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO
MARANHÃO
Sistemas de Informação
Algoritmos e Estruturas de Dados – Prof. Gentil Cutrim Serra Jr
3ª Avaliação – 13/12/2022

`recursao(27) = recursao(8)`

Como $8 \leq 10$, então `recursao(8) = 2 * 8 = 16`. Por fim,

`recursao(27) = 16`

4) [2pt] Escreva uma função em linguagem C que receba um ponteiro para uma árvore binária **t** e um ponteiro para um nó dessa árvore (subárvore **s**) e retorne o somatório do conteúdo (números inteiros) de todos os nodos da subárvore **s**. Você poderá usar mais de uma função. Na estrutura do nodo, considere que os valores encontram-se no campo **valor**. Por exemplo, se a função receber como parâmetro a árvore representada na Ilustração 1 e um endereço que aponte para o nodo com conteúdo 6, a função deverá retornar o valor 18 (5+6+7). (2pt).

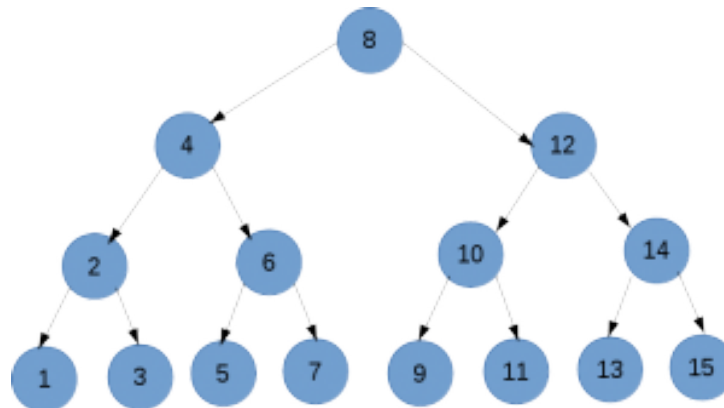


Ilustração 1

```
int procura (tree t, tree s) {  
    int soma=0;  
    if (t != NULL) {  
        if (t == s)  
            soma = somaConteudoNos(s);  
        else  
            if ((soma = procura(t->esq, s)) == 0)  
                soma = procura(t->dir, s);  
    }  
    return soma;  
}
```



INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO
MARANHÃO

Sistemas de Informação

Algoritmos e Estruturas de Dados – Prof. Gentil Cutrim Serra Jr

3ª Avaliação – 13/12/2022

```
int somaConteudoNos(tree t) {  
    if (t != NULL)  
        return (t->valor + somaConteudoNos(t->esq) +  
somaConteudoNos(t->dir));  
    else  
        return 0;  
}
```

5) [2pt] Escreva um código em C que receba como parâmetros um ponteiro para uma árvore binária e soma todos os conteúdos `t->info` dos nós folhas dessa árvore.

```
int somaFolhas(tree t) {  
    if (t != NULL)  
        if ((t->esq == NULL) && (t->dir == NULL)) //é folha  
            return (t->valor)  
        else  
            return (somaConteudoNos(t->esq) + somaConteudoNos(t->dir));  
    else  
        return 0;  
}
```

6) [2pt] Faça uma função que receba uma árvore `t`, conforme o formato lógico da ilustração 2, e imprima `t->info` de todos os *filhos mais novos (caçulas)*, caso se tenha mais de um irmão (Pedro, Beth).

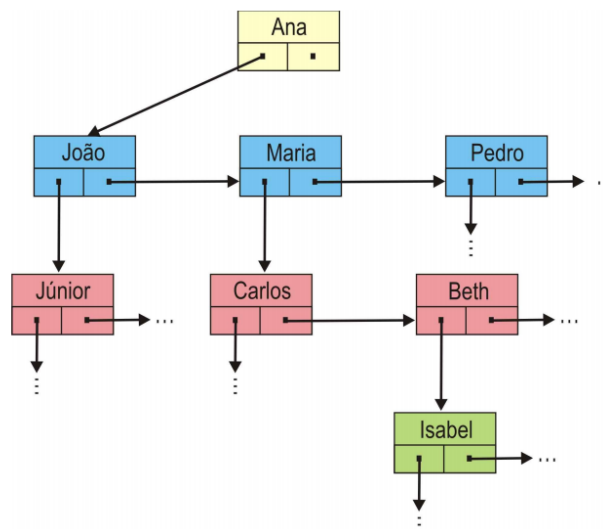


Ilustração 2



INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO
MARANHÃO

Sistemas de Informação

Algoritmos e Estruturas de Dados – Prof. Gentil Cutrim Serra Jr

3ª Avaliação – 13/12/2022

```
void imprime_caculas (tree t) {  
    tree aux;  
    if (t!=NULL) {  
        if (t->esq!=NULL) //tem filho  
            if (t->esq->dir!=NULL) { //o filho tem pelo menos 1  
irmão  
                aux = t->esq->dir;  
                while (aux!=NULL)  
                    aux = aux->dir;  
                printf("%d", t->esq);  
            }  
            imprime_caculas(t->esq);  
        }  
        imprime_caculas(t->dir);  
    }  
}
```

7) [QUESTÃO EXTRA] Dada a árvore representada na Ilustração 3, desenhe a sua árvore binária correspondente (1pt).

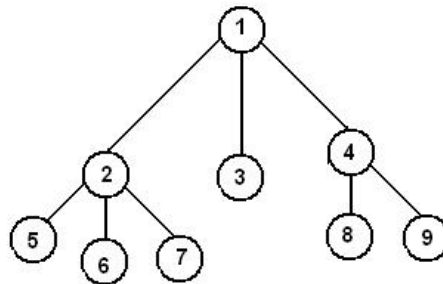


Ilustração 3



**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO
MARANHÃO**

Sistemas de Informação

Algoritmos e Estruturas de Dados – Prof. Gentil Cutrim Serra Jr

3ª Avaliação – 13/12/2022

