

Exercício: Algoritmos e Estrutura de Dados I
Universidade Federal do ABC

Árvore de RAs

O professor ABC guarda os RAs (número de registro do aluno) de todos que tiraram nota final A. Sempre que um aluno tira A ao final do quadrimestre, o RA dele é armazenado. Algumas vezes, o professor procura na lista se um determinado RA já está na lista.

Recentemente, um aluno sugeriu que ele implementasse esse sistema utilizando uma **árvore binária de busca**. O aluno argumentou que assim a busca e a inserção de novos RAs ficariam mais rápidas.

Você pode ajudar o professor ABC a implementar este programa?

Exercício:

Escreva um programa que receba uma sequência de operações (inserção e obter profundidade) e, quando for feita uma **operação de obter profundidade**, o programa deve imprimir a profundidade do nó na **árvore binária de busca**.

Código

Escreva o programa em linguagem C e salve o código no arquivo: **arvore_ra.c**

O programa deverá utilizar uma árvore binária de busca.

Entrada

O sistema receberá diversas entradas no formato [Operação] [RA], em que [Operação] pode ser I (inserção) ou P (obter profundidade). A operação é do tipo **char** e o RA é do tipo **int**. **O programa deve parar quando receber a entrada 'E 0'.**

Na operação **I** (inserção), o RA deverá ser adicionado na árvore binária de busca. Na operação **P** (obter profundidade), será apresentada a profundidade do nó com o RA informado.

Observação: Um RA pode ser qualquer número inteiro, inclusive zero e negativos.

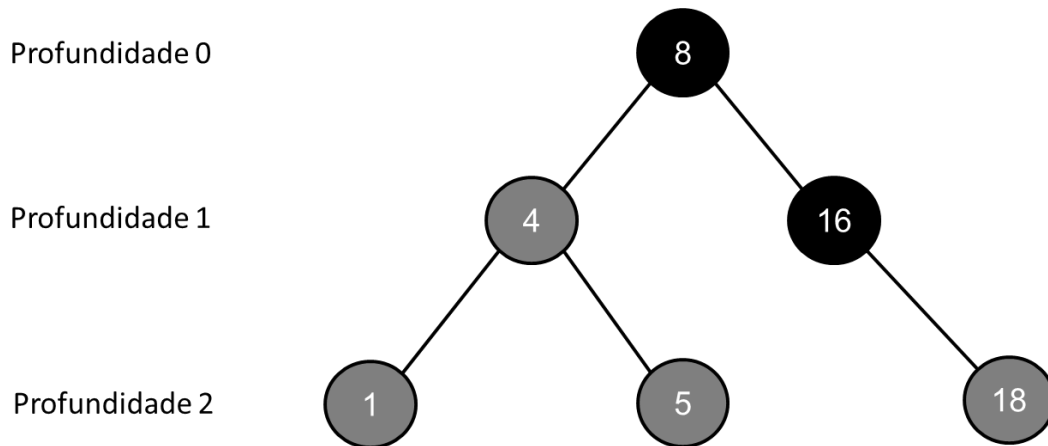
Saída

Somente quando ocorrer uma operação **P** (profundidade), o programa imprimirá a profundidade, de acordo com o seguinte formato:

$P=[P_No]$

Em que $[P_No]$ é a profundidade do nó na árvore. Considere profundidade = -1 caso RA informado não exista na árvore.

Por exemplo, veja a figura abaixo:



A árvore da figura anterior possui os RAs inseridos na ordem 8, 4, 16, 1, 5, 18.

Neste exemplo, o RA=16 está na profundidade 1. A saída para P 16 será (profundidade do RA 16):

$P=2$

Importante: libere a árvore da memória ao final do programa. Utilize o percurso pós-ordem e imprima os nós sendo removidos. Por exemplo, na árvore acima serão imprimidos os RA na seguinte ordem: 1 5 4 18 16 8

Exemplo 1 (as saídas estão em verde)

I 8
I 4
I 16
I 1
I 5
I 18
P 8
P=0
P 18
P=2
P 1
P=2
P 5
P=2
P 16
P=1
P 78
P=-1
P -9
P=-1
P 15
P=-1
I 15
P 15
P=2
E 0
1 5 4 15 18 16 8

Exemplo 2 (as saídas estão em verde)

I 1
I 4
I 5
I 8
I 16
I 18
P 8
P=3
P 18
P=5
P 1
P=0
P 5
P=2
P 16
P=4
P 78
P=-1
P -9
P=-1
P 15
P=-1
I 15
P 15
P=5
E 0
15 18 16 8 5 4 1