

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

Lista 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 adicionado na lista. Como o número de alunos que tiram A é imprevisível, o professor escolheu usar uma **lista simplesmente ligada** em seu sistema. Algumas vezes, o professor procura na lista se um determinado RA já está na lista. Por isso, o professor mantém essa lista ordenada (de forma crescente) por RA.

Recentemente, um aluno sugeriu que ele implementasse essa lista utilizando uma **árvore binária**. O aluno argumentou que assim a busca e a inserção de novos RAs ficariam mais rápidas. Desconfiado, o professor resolveu implementar um sistema que guarda duas cópias dos RAs: uma em **lista simplesmente ligada** e outra em uma **árvore binária de busca**.

O professor resolveu então comparar a quantidade de comparações que são feitas nas duas estruturas para averiguar se o aluno está certo. Você pode ajudar o professor ABC a implementar este programa?

Exercício:

Escreva um programa que receba uma sequência de operações (inserção e busca) e, quando for feita uma **operação de busca**, o programa deve imprimir a quantidade de comparações realizadas na **lista simplesmente ligada** e na **árvore binária de busca**.

Código

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

O programa deverá utilizar uma lista simplesmente ligada e uma árvore binária de busca.

Entrada

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

Na operação **I** (inserção), o RA deverá ser adicionado na lista simplesmente ligada e na árvore binária de busca. Na operação **B** (busca), será realizada uma busca do RA informado na lista ligada e na árvore.

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

Importante: a lista simplesmente ligada é ordenada, portanto, ela mantém os RAs ordenados de forma crescente.

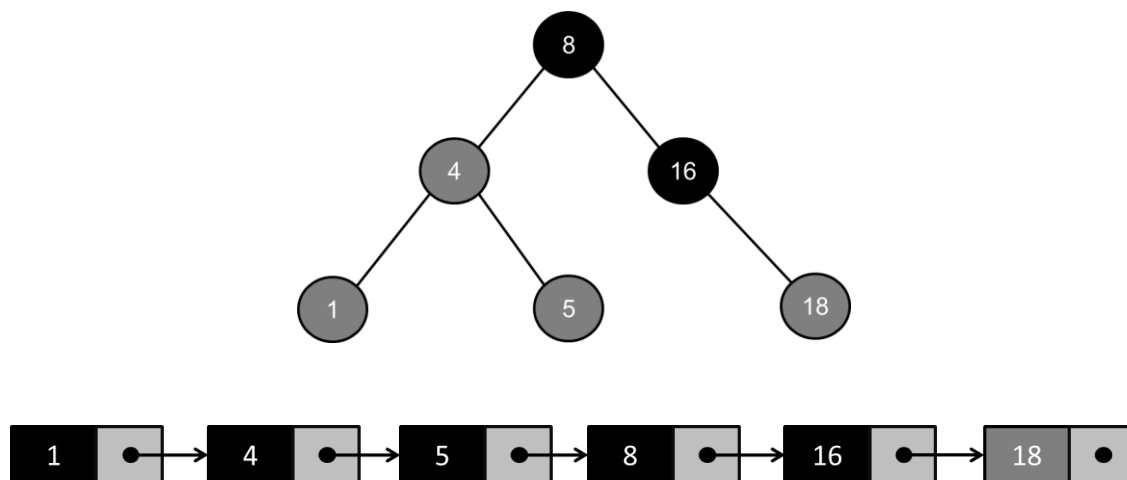
Saída

Somente quando ocorrer uma operação **B** (busca), o programa imprimirá a quantidade de comparações realizadas na operação de busca na lista ligada e na árvore, de acordo com o seguinte formato:

L=[NLista] A=[NArvore]

[NLista] é a quantidade de comparações realizadas até o fim da **busca** na lista ligada e [NArvore] é quantidade de comparações realizadas até o fim da **busca** na árvore.

Por exemplo, veja as figuras abaixo:



As duas estruturas da figura anterior possuem os mesmos RAs, inseridos na ordem 8, 4, 16, 1, 5, 18. Observe que a lista ligada manteve os RAs ordenados de forma crescente.

Neste exemplo, a busca pelo RA=16 resulta em cinco comparações na lista simplesmente ligada (nós marcados em preto) e duas comparações na árvore binária de busca (nós marcados em preto). A saída para B 16 será (buscar o RA 16):

L=5 A=2

Exemplo 1 (as saídas estão em verde):

I 8
I 4
I 16
I 1
I 5
I 18
B 8
L=4 A=1
B 18
L=6 A=3
B 1
L=1 A=3
B 5
L=3 A=3
B 16
L=5 A=2
B 78
L=6 A=3
B -9
L=1 A=3
B 15
L=5 A=2
I 15
B 15
L=5 A=3
P 0

Exemplo 2 (as saídas estão em verde):

I 1
I 4
I 5
I 8
I 16
I 18
B 8
L=4 A=4
B 18
L=6 A=6
B 1
L=1 A=1
B 5
L=3 A=3

B 16

L=5 A=5

B 78

L=6 A=6

B -9

L=1 A=1

B 15

L=5 A=5

I 15

B 15

L=5 A=6

P 0