

# MC202 (Estruturas de Dados) - 1s2021

## Tarefa 08: Árvores binárias de busca

Profº Ricardo Dahab

Assistente: Elisa Dell'Arriva

Assistente: Jônatas Trabuco Belotti

Instituto de Computação - UNICAMP

### Sobre a Tarefa

O objetivo desta tarefa é que o(a) aluno(a) se familiarize com a criação e manipulação de árvores binárias de busca. O exercício consiste em fazer a leitura de uma sequência de operações de inserção e de remoção e, ao final, imprimir a árvore resultante.

### Árvores binárias de busca

Uma árvore binária é dita *árvore binária de busca* se, para todo nó  $n$ , a subárvore à esquerda de  $n$  contém somente números com valores menores que o valor de  $n$  e a subárvore à direita de  $n$  contém somente nós com valores maiores que o valor de  $n$ .

Nesta tarefa, a *struct* que representa um nó deve, obrigatoriamente, ser a seguinte.

```
struct no {
    int valor;
    struct no *esquerda, *direita;
}
```

Note que a estrutura de um nó NÃO tem apontador para o seu nó pai. Nesta tarefa, não é permitido que cada nó tenha um apontador para seu nó pai. A operação de remoção deve ser feita conforme material apresentado em aula (ou seja, removendo-se o sucessor). Em todos os casos de testes, só são removidos valores que foram anteriormente inseridos, isto é, não há ocorrência de remoção de um valor não existente na árvore.

### Formato da entrada e saída

Cada linha da entrada consiste em um operador e um valor separados por espaços, no seguinte formato.

<operador> <valor do nó>

O **operador** pode ser I, indicando inserção, R, indicando remoção, ou F, indicando o fim da sequência de operações. O **valor do nó** serão números inteiros. A saída consiste em três linhas. A primeira linha deve ser a árvore resultante impressa em percurso pré-ordem; a segunda linha deve ser a árvore resultante impressa em percurso in-ordem e, finalmente, a terceira linha deve ser a árvore resultante em percurso pós-ordem. Cada uma das três linhas deve ser seguida de uma quebra de linha. Quando a árvore resultante for vazia, deve-se imprimir a mensagem "ARVORE RESULTANTE VAZIA", com uma quebra de linha ao final.

### Exemplo 1.

Entrada:

```
I 2
I 0
I 2
R 0
I 0
F
```

Saída:

```
2 0 2
0 2 2
0 2 2
```

### Exemplo 2.

Entrada:

```
I 2940
I 1177
I 8089
I 1910
I 2908
I -208
I 4807
I 9384
I 7674
R 7674
R 1910
R 2908
F
```

Saída:

```
2940 1177 -208 8089 4807 9384
-208 1177 2940 4807 8089 9384
-208 1177 4807 9384 8089 2940
```

### Exemplo 3.

Entrada:

```
I 202
R 202
F
```

Saída:

```
ARVORE RESULTANTE VAZIA
```

## Importante

- É proibido o uso de apontador para o pai de cada nó dentro da estrutura que representa um nó. O não cumprimento dessa regra implicará em nota zero nesta tarefa.
- É obrigatório utilizar a *struct* descrita neste enunciado para representar um nó. O não cumprimento dessa regra implicará em nota zero nesta tarefa.
- É obrigatório liberar toda memória alocada dinamicamente. Para fazer essa verificação, recomendamos o uso da ferramenta *valgrind*. Caso haja memória não liberada, serão descontados dois pontos da nota final desta tarefa.

## Alguns avisos e lembretes

- A página da disciplina no SuSy é <https://susy.ic.unicamp.br:9999/mc202abc>.
- Para submeter, utilizem somente os dígitos numéricos do RA e a senha da DAC.
- Espere, no mínimo, 60 segundos entre uma submissão e outra no SuSy, para evitar problemas na entrega.
- O número máximo de submissões é 20.
- Esta tarefa tem peso 1 (um).
- O prazo de entrega desta tarefa é 21/06/2021, às 23h59. Serão aceitas submissões atrasadas até o dia 24/06/2021 e serão aplicados os descontos de nota estabelecidos conforme o número de dias em atraso.