

Dredd - Juiz Online

Principal

Perfil

Minhas Provas

Sair

Minutos
Restantes:
14702

Usuário:
Lucas Antonio
Lopes Neves

Notas:
Q1: ?
Q2: ?
Total: 0

Exercícios sobre Árvores

Prova Aberta Até: 27/10/2018 03:00:00

Número Máximo de Tentativas: 6

Atenuação da Nota por Tentativa: 2%

Instruções para a prova: Exercícios sobre árvores. Pode ser acessada de casa.

Questão 1: ABB - Implementação Básica

Implemente uma Árvore Binária de Busca (ABB) com operações para inserir, remover e escrever os elementos de duas formas (em ordem e pré-ordem). A árvore criada deve ser capaz de armazenar um único tipo de informação (chave). As chaves no programa serão números inteiros, porém, quando mais independente for a classe, melhor. A estratégia a respeito de como lidar com chaves repetidas não é importante.

As operações para escrever elementos devem sempre escrever a chave, uma barra e o nível na árvore em que a chave está. Isso vai ajudar a determinar a estrutura da árvore ao testar o programa. Não devem ser colocados espaços antes nem depois da barra.

Caso tentem remover uma chave que não está na árvore, o programa deverá escrever "ERRO" (letras maiúsculas, sem as aspas) na saída padrão. A operação de escrita deve estar na função principal (programa) e não em algum método.

A estratégia de remoção de nó com dois filhos deve ser a de substituir pelo sucessor.

O programa deverá ler comandos identificados por letras minúsculas e seus parâmetros (quando necessário). Os comandos possíveis devem ser:

- A letra `i`, seguida de uma chave para `inserir` uma chave na árvore.
- A letra `r`, seguida de uma chave para `remover` uma chave da árvore.
- A letra `o` para escrever os elementos em `ordem`, no formato descrito acima.
- A letra `p` para escrever os elementos em `pré-ordem`, no formato descrito acima.
- A letra `f` para `finalizar` a execução do programa.

Entradas:

Minutos
Restantes:
14702

Usuário:
Lucas Antonio
Lopes Neves

Notas:
Q1: ?
Q2: ?
Total: 0

Uma sequência de comandos, conforme especificado acima.

Saídas:

Somente os comandos para escrever produzem saída, conforme formato explicado acima.

Exemplo de Entrada:

```
i 3
i 4
i 2
i 5
i 1
r 3
o
p
q
```

Exemplo de Saída:

```
1/2 2/1 4/0 5/1
4/0 2/1 1/2 5/1
```

Peso: 1

Nova Resposta: _____

Selecione o arquivo com o código fonte do programa que resolve o problema para enviá-lo.

Choose File No file chosen

Enviar Resposta

Questão 2: Árvore binária (ABB) - árvore sem nó

Existem muitas formas de implementar uma estrutura de dados qualquer. A linguagem de programação usada é um dos fatores que mais influência as características de implementação. As linguagens que escondem ponteiros e usam *coleta de lixo* para gerenciar o uso da memória favorecem a implementação de árvores sem a classe auxiliar "nó".

Linguagens sem *coleta de lixo* não favorecem esse tipo de implementação, mas isso não quer dizer que não podemos fazer uma.

Implemente uma ABB sem usar uma classe auxiliar. Para agilizar seu desenvolvimento, use [este programa](#) que já tem atributos e métodos projetados, além de um programa com interface para inserir, buscar, remover e escrever a árvore. Os métodos auxiliares no código são sugestão de implementação. O programa lê comandos numéricos e seus argumentos para as várias operações possíveis.

Minutos Restantes:
14702

Usuário:
Lucas Antonio
Lopes Neves

Notas:
Q1: ?
Q2: ?
Total: 0

Entradas:

Cada comando é um número inteiro identificando o comando seguido dos parâmetros necessários para executar o comando, se houverem. Os códigos de comandos são:

- O número 0 para encerrar a execução do programa.
- O número 1 para inserir chave (número inteiro) e valor (número inteiro) na árvore
- O número 2 para remover dado da árvore, seguido da chave (número inteiro) que deve ser removida.
- O número 3 para buscar na árvore, seguido da chave consultada. Este comando produz uma saída que é o valor associado.
- O número 4 para escrever todas chaves e seus respectivos valores num formato de texto com parênteses.
- O número 5 para escrever os nós da árvore nível a nível.

Saídas:

Cada comando tem sua saída específica. Veja os exemplos abaixo.

Exemplo de entrada e saída juntos:

```
1 5 50
1 3 30
1 2 20
4
(5/50 (3/30 (2/20 () ()) ()) ())
1 6 60
5
[5/50]
[3/30][6/60]
[2/20][][][]
[][]
2 5
5
[6/60]
[3/30][]
[2/20][]
[][]
2 4
Impossível remover. A chave não existe.
2 3
4
(6/60 (2/20 () ()) ())
2 2
2 6
5
[]
0
```

Peso: 1

Nova Resposta: _____

Selecione o arquivo com o código fonte do programa que resolve o problema para enviá-lo.

No file chosen

**Minutos
Restantes:**
14702

Usuário:
Lucas Antonio
Lopes Neves

Notas:
Q1: ?
Q2: ?
Total: 0



Desenvolvido por Bruno
Schneider a partir do programa
original (Algod) de Renato R.
R. de Oliveira.

