Exercícios sobre Árvores

Prova Aberta Até: 01/12/2018 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:

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
f
```

Exemplo de Saída:

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

Peso: 1

Última tentativa realizada em: 25/10/2018 10:08:18

Tentativas: 3 de 6

Nota (0 a 100): 96.1

Status ou Justificativa de Nota: Nenhum erro encontrado.

Ver Código da Última Tentativa

Nova Resposta:

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

Browse... Enviar Resposta

Minutos Restantes: 1997

Usuário: Lucas Antonio Lopes Neves

Notas:
Q1: 96.1
Q2: ?
Q3: 100
Q4: 98
Q5: ?
Q6: ?
Q7: ?
Q8: 0
Q9: ?
Q10: ?
Q11: ?
Total: 27

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.

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.

Browse... Enviar Resposta

Questão 3: Árvore binária (ABB) - Árvore de Palavras

Considere um arquivo texto preenchido de nome `entrada.txt`. Esse arquivo deve ser o parâmetro de entrada para o seu sistema que deve:

- armazenar cada palavra do texto em um nó de uma árvore binária de busca
- armazenar, para cada palavra, a posição em que ela aparece no texto
- permitir a busca por uma determinada palavra. Caso a palavra esteja presente na árvore, o programa deverá imprimir as posições em que se encontra. Caso a palavra não seja encontrada, o programa deve imprimir -1

Exemplo:

E agora, Jose?

A festa acabou,

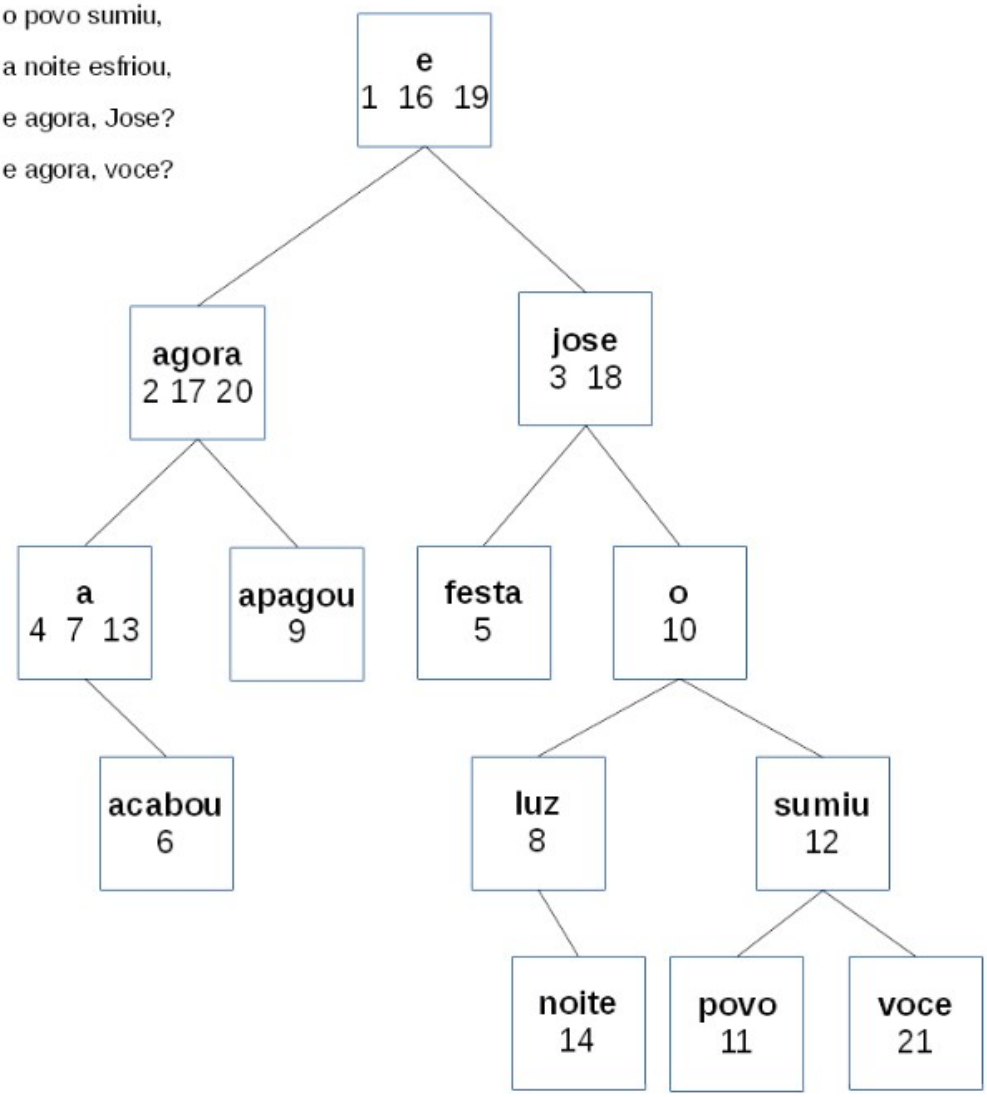
a luz apagou,

o povo sumiu,

a noite esfriou,

e agora, Jose?

e agora, voce?



Neste exemplo a palavra **e** aparece nas posições 1, 16 e 19, ou seja, ela aparece mais de uma vez no texto, mas deve ser representada por apenas um nó na árvore. As posições das repetições devem ser armazenadas no mesmo nó.

Observação: Considere que o texto presente no arquivo já passou por um tratamento, assim já foram removidos acentos e pontuação além de todas as letras serem minúsculas.

Entradas:

1. Arquivo de entrada

2. Palavra a ser buscada

Saídas:

1. Posições da palavra

Exemplo de Entrada:

No arquivo:

```
e agora jose
a festa acabou
a luz apagou
o povo sumiu
a noite esfriou
e agora jose
e agora voce
```

Na entrada padrão:

```
agora
```

Exemplo de Saída:

```
2 17 20
```

Minutos Restantes: 1997

Usuário: Lucas Antonio Lopes Neves

Notas:
Q1: 96.1
Q2: ?
Q3: 100
Q4: 98
Q5: ?
Q6: ?
Q7: ?
Q8: 0
Q9: ?
Q10: ?
Q11: ?
Total: 27

Peso: 1

Última tentativa realizada em: 07/11/2018 10:47:26

Tentativas: 1 de 6

Nota (0 a 100): 100

Status ou Justificativa de Nota: Nenhum erro encontrado.

Ver Código da Última Tentativa

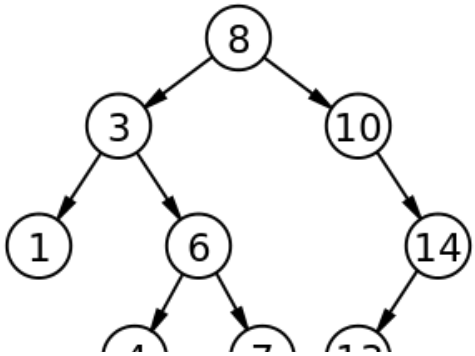
Nova Resposta:

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

Browse... Enviar Resposta

Questão 4: Árvore binária (ABB) - sucessor

Implemente um método para encontrar o sucessor de uma chave numa árvore binária. Suponha que a chave pertence à árvore. É importante que haja um mínimo de eficiência: buscar a chave (já pronto) e depois caminhar até o sucessor, sem processar outros nós.



Use esta implementação que já tem prontos métodos para inserir chave, buscar chave e escrever a árvore, além de uma interface que ativa os métodos implementados. A interface não deve ser alterada.

Antes de implementar, planeje. Note que um nó que não tem filho à direita tem um sucessor que não está numa subárvore. Na árvore da figura, por exemplo, o sucessor de 7 está na raiz.

Entradas:

O programa lê comandos e os executa. Os sucessores são encontrados chamando repetidamente o método que encontra um sucessor. Basta usar a interface providenciada. Os comandos são:

- f - para finalizar a execução do programa
- i - para inserir uma chave, seguido da chave (inteiro)
- e - para escrever o conteúdo da árvore, nível a nível para facilitar o entendimento da estrutura
- s - para escrever todos os sucessores de uma chave, seguido da chave (inteiro)

Saídas:

Cada comando produz uma saída específica. Apenas mantenha as saídas já implementadas.

Exemplo de entrada e saída juntos:

```
i 300
i 20
i 430
i 12
i 360
i 451
s 20
300 360 430 451
f
```

Peso: 1

Última tentativa realizada em: 18/10/2018 17:51:11

Tentativas: 2 de 6

Nota (0 a 100): 98

Status ou Justificativa de Nota: Nenhum erro encontrado.

Ver Código da Última Tentativa

Nova Resposta:

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

Browse... Enviar Resposta

Minutos Restantes: 1997

Usuário: Lucas Antonio Lopes Neves

Notas:
Q1: 96.1
Q2: ?
Q3: 100
Q4: 98
Q5: ?
Q6: ?
Q7: ?
Q8: 0
Q9: ?
Q10: ?
Q11: ?
Total: 27

Questão 5: AVL - inserção

Implemente uma classe para representar árvores AVL, que contém operações para inserir elementos e para escrever a estrutura da árvore. O código fornecido contém a implementação para escrever a estrutura e também uma sugestão de atributos e métodos necessários. O atributo `altura` da classe nó é obrigatório e é usado no método de escrita. Também já está implementado o programa que usa a classe AVL.

Certifique-se de implementar corretamente todos os 4 casos de balanceamento ao invés de apenas testar o exemplo de E/S dado neste enunciado.

Entradas:

O programa lê códigos de comandos a executar. Sempre um caractere identificando o comando, seguido dos parâmetros necessários para executar o comando, se houverem. Os códigos de comandos são:

- i: para inserir um elemento (string / número real) à árvore;
- e: para escrever a estrutura da árvore.

Saídas:

Somente o comando para escrever produz saída no formato especificado, que já está implementado.

Exemplo de Entrada:

```
i andre 19
e
i antonio 29
e
i carla 28
e
i diego 52
e
i eduardo 63
e
f
```

Exemplo de Saída:

```
[1:andre/19]
[] []

[2:andre/19]
[] [1:antonio/29]
[] []

[2:antonio/29]
[1:andre/19] [1:carla/28]
[] [] [] []

[3:antonio/29]
[1:andre/19] [2:carla/28]
[] [] [] [1:diego/52]
[] []

[3:antonio/29]
[1:andre/19] [2:diego/52]
[] [] [1:carla/28] [1:eduardo/63]
[] [] [] []
```

Peso: 1

Nova Resposta:

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

Browse... Enviar Resposta

Questão 6: AVL - implementação completa

Implemente uma classe para representar árvores AVL, que contém operações para inserir, buscar, remover e escrever a estrutura da árvore. O código fornecido contém a implementação para escrever a estrutura e também uma sugestão de atributos e métodos necessários. O atributo `altura` da classe nó é obrigatório e é usado no método de escrita. Também já está implementado o programa que usa a classe AVL.

Entradas:

O programa lê códigos de comandos a executar. Sempre um caractere identificando o comando, seguido dos parâmetros necessários para executar o comando, se houverem. Os códigos de comandos são:

- i: para inserir um elemento (string / número real) à árvore;
- b: para bscar um valor (número real) associado a uma chave (string);
- r: para remover um elemento, seguido da chave (string) do elemento;
- e: para escrever a estrutura da árvore.

Saídas:

Somente o comando para escrever produz saída no formato especificado, que já está implementado.

Exemplo de Entrada:

```
i andre 19
e
i antonio 29
e
i carla 28
e
i diego 52
e
i eduardo 63
e
r diego
e
r antonio
e
b eduardo
f
```

Exemplo de Saída:

```
[1:andre/19]
[] []

[2:andre/19]
[] [1:antonio/29]
```

```
[] []

[2:antonio/29]
[1:andre/19] [1:carla/28]
[] [] [] []

[3:antonio/29]
[1:andre/19] [2:carla/28]
[] [] [] [1:diego/52]
[] []

[3:antonio/29]
[1:andre/19] [2:diego/52]
[] [] [1:carla/28] [1:eduardo/63]
[] [] [] []

[3:antonio/29]
[1:andre/19] [2:eduardo/63]
[] [] [1:carla/28] []
[] []

[2:carla/28]
[1:andre/19] [1:eduardo/63]
[] [] [] []

63
```

Peso: 1

Nova Resposta:

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

Browse... Enviar Resposta

Questão 7: Árvore Rubro-Negra - inserção

Implemente uma classe para representar árvores rubro-negras, que contém operações para inserir elementos e para escrever a estrutura da árvore. O código fornecido contém a implementação para escrever a estrutura e também uma sugestão de atributos e métodos necessários. Também já está implementado o programa que usa a classe.

Certifique-se de implementar corretamente todos os cinco casos de arrumação da árvore ao invés de apenas testar o exemplo de E/S dado neste enunciado.

Entradas:

O programa lê códigos de comandos a executar. Sempre um caractere identificando o comando, seguido dos parâmetros necessários para executar o comando, se houverem. Os códigos de comandos são:

- i: para inserir um elemento (chave/valor) à árvore (chaves são números naturais, valores são texto);
- e: para escrever a estrutura da árvore.

Saídas:

Somente o comando para escrever produz saída no formato especificado, que já está implementado.

Exemplo de Entrada:

```
i 1 um
e
i 2 dois
e
i 3 tres
e
i 4 quatro
e
i 5 cinco
e
f
```

Exemplo de Saída:

```
[P:1/um]
NIL NIL

[P:1/um]
NIL [V:2/dois]
NIL NIL

[P:2/dois]
[V:1/um] [V:3/tres]
NIL NIL NIL NIL

[P:2/dois]
[P:1/um] [P:3/tres]
NIL NIL NIL [V:4/quatro]
NIL NIL

[P:2/dois]
[P:1/um] [P:4/quatro]
NIL NIL [V:3/tres] [V:5/cinco]
NIL NIL NIL NIL
```

Peso: 1

Nova Resposta:

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

Browse... Enviar Resposta

Questão 8: Árvore 2-3 - inserção

Implemente uma classe para representar árvores 2-3, que contém operações para inserir elementos e para escrever a estrutura da árvore. O código fornecido contém a implementação para escrever a estrutura e também uma sugestão de atributos e métodos necessários. Também já está implementado o programa que usa a classe Arvore23.

Na sugestão de implementação passada, cada nó criado ganha um identificador único (inteiro positivo). Ele ajuda a depurar o programa e é usado no método de escrita. Não é permitido alterar essa estratégia.

Outras decisões de implementação são: a) Um nó pode armazenar temporariamente 3 chaves e 4 filhos. Isto deixa a implementação mais parecida com os diagramas apresentados no material teórico. b) O filho mais à esquerda (índice 0) indica se um nó é folha. c) Os nós tem atributo pai. Essas estratégias não são obrigatórias.

Entradas:

O programa lê códigos de comandos a executar. Sempre um caractere identificando o comando, seguido dos parâmetros necessários para executar o comando, se houverem. Os códigos de comandos são:

- i: para inserir um elemento (número inteiro) à árvore;
- e: para escrever a estrutura da árvore.

Saídas:

Somente o comando para escrever produz saída. Os nós são escritos na ordem que estão na estrutura, nível a nível, no formato **[identificador|chaves|pai;filhos]**.

Exemplo de Entrada:

```
i 40
i 12
i 68
i 36
i 38
i 60
i 48
e
i 55
i 50
e
i 62
i 65
e
f
```

Exemplo de Saída:

```
[7|40|0;3,6]
[3|36|7;1,4] [6|60|7;2,5]
[1|12|3;] [4|38|3;] [2|48|6;] [5|68|6;]

[7|40|0;3,6]
[3|36|7;1,4] [6|50,60|7;2,8,5]
[1|12|3;] [4|38|3;] [2|48|6;] [8|55|6;] [5|68|6;]

[7|40,60|0;3,6,10]
[3|36|7;1,4] [6|50|7;2,8] [10|65|7;5,9]
[1|12|3;] [4|38|3;] [2|48|6;] [8|55|6;] [5|62|10;] [9|68|10;]
```

Peso: 1

Última tentativa realizada em: 22/11/2018 11:44:32

Tentativas: 1 de 6

Nota (0 a 100): 0

Status ou Justificativa de Nota: O programa não compila.

Ver Código da Última Tentativa

Nova Resposta:

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

Browse... Enviar Resposta

Minutos Restantes: 1997

Usuário: Lucas Antonio Lopes Neves

Notas:
Q1: 96.1
Q2: ?
Q3: 100
Q4: 98
Q5: ?
Q6: ?
Q7: ?
Q8: 0
Q9: ?
Q10: ?
Q11: ?
Total: 27

Questão 9: Inserção em Arvore 2-3-4

Implemente uma classe para representar árvores 2-3-4, que contém operações para inserir elementos e para escrever a estrutura da árvore. O código fornecido contém a implementação para escrever a estrutura e também uma sugestão de atributos e métodos necessários. Já está implementado o programa que usa a classe bt234.

Certifique-se de implementar corretamente as situações de inserção (i.e., inserção em nós com menos de três valores e mais de três valores).

Entradas:

O programa lê códigos de comandos a executar. Sempre um caractere identificando o comando, seguido dos parâmetros necessários para executar o comando, se houverem. Os códigos de comandos são:

- i: para inserir um elemento (número real) à árvore;
- e: para escrever a estrutura da árvore.
- q: para sair.

Saídas:

Somente o comando para escrever produz saída no formato especificado, que já está implementado.

Exemplo de Entrada:

```
i 10.5
i 5
i 8
e
i 25
i 20
i 12
e
q
```

Exemplo de Saída:

```
5/0 8/0 10.5/0
5/1 8/0 10.5/1 12/1 20/0 25/1
```

Peso: 1

Nova Resposta:

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

Browse... Enviar Resposta

Minutos Restantes: 1997

Usuário: Lucas Antonio Lopes Neves

Notas:
Q1: 96.1
Q2: ?
Q3: 100
Q4: 98
Q5: ?
Q6: ?
Q7: ?
Q8: 0
Q9: ?
Q10: ?
Q11: ?
Total: 27

Questão 10: Minimo e Maximo em Arvore 2-3-4

Implemente uma classe para representar árvores 2-3-4, que contém operações para busca de minimo e máximo elementos e para escrever a estrutura da árvore. O código fornecido contém a implementação para escrever a estrutura e também uma sugestão de atributos e métodos necessários. Já está implementado o programa que usa a classe bt234.

Entradas:

O programa lê códigos de comandos a executar. Sempre um caractere identificando o comando, seguido dos parâmetros necessários para executar o comando, se houverem. Os códigos de comandos são:

- i: para inserir um elemento (número real) à árvore;
- h: para buscar e escrever o maior elemento na árvore.
- l: para buscar e escrever o menor elemento na árvore.
- e: para escrever a estrutura da árvore.
- q: para sair.

Saídas:

Somente o comando para escrever produz saída no formato especificado, que já está implementado.

Exemplo de Entrada:

```
i 10.5
h
l
i 5
i 8
i 25
i 20
i 12
e
h
l
q
```

Exemplo de Saída:

```
10.5
10.5
5/1 8/0 10.5/1 12/1 20/0 25/1
25
5
```

Peso: 1

Nova Resposta:

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

Browse... Enviar Resposta

Minutos Restantes: 1997

Usuário: Lucas Antonio Lopes Neves

Notas:
Q1: 96.1
Q2: ?
Q3: 100
Q4: 98
Q5: ?
Q6: ?
Q7: ?
Q8: 0
Q9: ?
Q10: ?
Q11: ?
Total: 27

Questão 11: Árvore B em RAM - implementação básica

Faça uma implementação de árvore B em memória (RAM). Use esta implementação, que já tem um método para escrever a árvore nível a nível para facilitar a depuração. O grau (ordem) da árvore deve ser definido na hora de criar uma árvore (grau > 1) e deve ser passado para o construtor. A definição de grau usada neste exercício é que o grau é o número mínimo de filhos de um nó (exceto a raiz) e o número máximo de filhos é *2*grau*.

Sua implementação deve ter construtores, destrutores e um método para inserção na árvore. A inserção deve ser preemptiva, ou seja, dividir qualquer nó cheio no caminho da inserção até uma folha. Decidir quais métodos são necessários e quais parâmetros de cada método também é uma parte deste exercício.

Entradas:

A implementação dada tem um programa que lê comandos e os executa. Basta usar a interface providenciada. O programa lê o grau da árvore e depois vários comandos. Os comandos são:

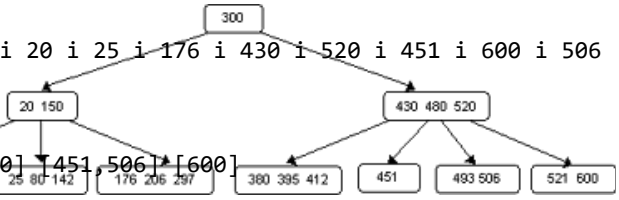
- f - para finalizar a execução do programa
- i - para inserir uma chave, seguido da chave (número inteiro)
- e - para escrever o conteúdo da árvore, nível a nível.

Saídas:

Cada comando produz uma saída específica. Apenas mantenha as saídas já implementadas.

Exemplo de Entrada e saída juntos:

```
2
i 12 i 300 i 360 i 20 i 25 i 176 i 430 i 520 i 451 i 600 i 506
e
[300]
[20] [430,520]
[12] [25,176] [360] [451,506] [600]
i 480 i 150 i 206
e
[300]
[20,150] [430,520]
[12] [25] [176,206] [360] [451,480,506] [600]
i 142 i 80 i 297
e
[300]
[20,150] [430,520]
[12] [25,80,142] [176,206,297] [360] [451,480,506] [600]
i 412 i 395
e
[300]
[20,150] [430,520]
[12] [25,80,142] [176,206,297] [360,395,412] [451,480,506] [600]
i 521 i 493
e
[300]
[20,150] [430,480,520]
```



Árvore criada com os valores de entrada do exemplo

```
[12] [25,80,142] [176,206,297] [360,395,412] [451] [493,506] [521,600]
f
```

Peso: 1

Nova Resposta:

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

Browse... Enviar Resposta

Minutos Restantes: 1997

Usuário: Lucas Antonio Lopes Neves

Notas:
Q1: 96.1
Q2: ?
Q3: 100
Q4: 98
Q5: ?
Q6: ?
Q7: ?
Q8: 0
Q9: ?
Q10: ?
Q11: ?
Total: 27