

Universidade de São Paulo – USP
Instituto de Ciências Matemáticas e de Computação – ICMC
Departamento de Ciências de Computação – SCC

SCC0202 - Algoritmos e Estruturas de Dados I

Responsável: Prof. Gustavo Batista
gbatista at icmc usp br

Estagiário PAE : Igor Bueno Corrêa
igorbc at icmc usp br

Projeto II – Árvores Balanceadas
Data de entrega: 14/12

Problema

Neste projeto você deve implementar uma árvore balanceada. Faça uma opção por Árvore AVL ou Vermelho-Preto. A sua implementação deve ter as seguintes operações:

- i. Inserção de um número inteiro na árvore balanceada em $O(\log n)$
- ii. Sucessor de uma chave em $O(\log n)$
- iii. Predecessor de uma chave $O(\log n)$
- iv. Máximo $O(\log n)$
- v. Mínimo $O(\log n)$
- vi. Impressão pré-ordem, em-ordem e pós-ordem em $O(n)$
- vii. Inicialização de uma árvore balanceada em $O(1)$
- viii. Desalocação da árvore em $O(n)$

Note que remoção de um único elemento não faz parte do conjunto obrigatório de operações, com a finalidade de simplificar o projeto.

Avaliação

1. 50% - Código fonte: implementação das operações, documentação e organização do código; utilização de módulos .c e .h; makefile;
2. 50% - Execução correta. Este teste será automatizado. Casos de teste serão submetidos ao programa e o retorno será comparado com o esperado. Os padrões de entrada e saída descritos a seguir devem ser estritamente respeitados;

Entrada

A entrada deve ser lida da entrada padrão e tem o seguinte formato:

A primeira linha contém o número de operações N . As próximas N linhas contém as operações, uma por linha.

As operações possuem o seguinte formato:

O X

No qual, O pode assumir 1 para inserção, 2 para sucessor, 3 para predecessor, 4 para máximo, 5 para mínimo, 6 para impressão pré-ordem, 7 para em-ordem e 8 para pós-ordem. Utilize as operações de inicialização e desalocação da árvore no início e final da execução, respectivamente. X é um valor inteiro e ocorre nas opções 1 (inserção), 2 (sucessor) e 3 (predecessor) apenas.

Saída

Para a operação 1 (inserção) não imprima nada na tela. Para as operações 2 (sucessor), 3 (predecessor), 4 (máximo) e 5 (mínimo) imprima um único número inteiro com a resposta em uma linha em separado. Para as operações 6 (impressão pré-ordem), 7 (em-ordem) e 8 (pós-odem), imprima uma sequência de números inteiros separados com um único espaço em branco entre números e em uma linha em separado. Para respostas a consultas inválidas, imprima a mensagem “erro” em uma linha em separado.

Exemplo de entrada

```
10
1 5
1 4
1 3
1 2
1 1
2 1
3 1
4
5
7
```

Exemplo de saída

```
2
erro
5
1
1 2 3 4 5
```

Implementação e entrega

O programa deve ser feito ou individualmente e deve ser implementado em C/C++/Java/Python. Todos os arquivos de código compactados devem ser entregues via TIDIA, fazendo upload dos arquivos na pasta “projeto 2” na ferramenta escaninho.