



**Universidade de São Paulo**

Instituto de Ciências Matemáticas e de Computação

Departamento de Ciências de Computação

SCC0202 – Algoritmos e Estruturas de Dados I

## Trabalho 3: Árvores Balanceadas

**Professores:** Dr. Adinovam H. M. Pimenta (adinovam@icmc.usp.br)

Dr. Gustavo E. A. P. A. Batista (gbatista@icmc.usp.br)

Dr. Marcelo G. Manzato (mmanzato@icmc.usp.br)

**Estagiário PAE:** Diego Silva (diego.fsilva@gmail.com)

**Monitores:** Felipe A. R. Dias (felipe.alegria.dias@usp.br)

Guilherme C. de Oliveira (guilherme.caixeta.oliveira@usp.br)

### 1. Descrição

Neste projeto você deve implementar a árvore Vermelho-Preto Clássica. 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.

### 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 **impressão em-ordem** e **8** para **impressão 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.

## 2. Exemplos de Entrada e Saída

### Exemplo de entrada

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

### Exemplo de saída

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

## 3. Instruções Complementares

- O trabalho deve ser desenvolvido individualmente por cada aluno, sendo este responsável por decidir quais estruturas de dados implementar, assim como as melhores opções de implementação;
- A implementação do trabalho será em linguagem C. Pode-se utilizar qualquer IDE/compilador para o desenvolvimento contanto que seja compilado adequadamente no compilador `gcc` usando as *flags* `-Wall`, `-ansi` e `-pedantic`.
- Compacte seu projeto e submeta o arquivo .zip no <http://run.codes>. O código da disciplina é: **BL9E**.

## 4. Critérios de Avaliação

- I. 50% - Código fonte: implementação das operações, documentação e organização do código; utilização de módulos .c e .h; makefile;
- II. 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.

## 5. Material Complementar

Para maiores informações sobre o funcionamento das árvores Preto-Vermelha, sugerimos a consulta ao seguinte material:

- Capítulo 13 do livro do Cormen<sup>1</sup>
- Capítulo 10 do livro do Goodrich<sup>2</sup>

---

<sup>1</sup> CORMEN, Thomas H. **Introduction to algorithms**. MIT press, 2009.

<sup>2</sup> GOODRICH, Michael; TAMASSIA, Roberto; MOUNT, David. **Data structures and algorithms in C++**. John Wiley & Sons, 2007.