

Aula 07 – Exercícios

Parte 1 – Escrita (individual – entregue via Formulário no Google Classroom)

Exercício 1: em seu caderno, faça, **passo a passo** (como feito em aula), a inserção dos seguintes elementos em uma Árvore Left-Leaning Red-Black, nesta ordem: 30, 51, 15, 44, 48, 22, 18, 5, 7, 41, 37.

Parte 2 – Codificação (individual OU em dupla OU em trio – entregue via Run Codes)

Exercício 2: escreva, em linguagem C pura (sem nenhum comando próprio de C++), um programa que leia um número inteiro representando uma opção do usuário. As ações a serem executadas dependerão da opção lida, sendo:

- Se a opção lida for **1**, seu programa deve ler um novo valor inteiro e inserir este valor em uma Árvore Red-Black.
 - Se a opção lida for **2**, seu programa deve ler um novo valor inteiro e pesquisá-lo na Árvore Red-Black. Se o valor pesquisado estiver presente na Árvore, imprima-o. Se o valor pesquisado não estiver presente na Árvore, imprima um “x” minúsculo como resposta. Pule uma linha após imprimir o número ou o “x”.
 - Se a opção lida for **3**, seu programa deve executar o percurso em pré-ordem na Árvore Red-Black. Imprima os valores dos elementos em uma única linha, separados por um único espaço. Pule uma linha ao finalizar o percurso.
 - Se a opção lida for **4**, seu programa deve executar o percurso em ordem (in-order) na Árvore Red-Black. Imprima os valores dos elementos em uma única linha, separados por um único espaço. Novamente, pule uma linha ao finalizar o percurso.
 - Se a opção lida for **5**, seu programa deve executar o percurso em pós-ordem na Árvore Red-Black. Imprima os valores dos elementos em uma única linha, separados por um único espaço. Pule uma linha ao finalizar o percurso.
 - Se a opção lida for **6**, seu programa deve ler um novo valor, pesquisar pelo nó que contém o valor lido e imprimir a cor deste nó (os casos de testes não irão trapacear seu programa pesquisando por um nó inexistente, mas testará apenas a cor para os nós presentes na árvore!). Se o nó for vermelho, imprima como resposta um “R” maiúsculo. Se o nó for preto, imprima um “B” maiúsculo. Pule uma linha após imprimir a letra correspondente.
 - Se a opção lida for **7**, seu programa deve finalizar. Note que seu programa encerra apenas nesta opção. Ao terminar de processar qualquer outra opção, o programa deve retornar ao loop inicial para ler a próxima ação que o usuário deseja executar.
-

Observação 1: este exercício está dividido em uma parte escrita e uma parte prática de codificação. A entrega só será válida se as duas partes forem entregues.

Observação 2: todos os alunos devem entregar a parte escrita e a parte prática individualmente. Não há problema em entregar o código 100% igual ao de sua dupla (ou trio), porém, cada aluno deve entregar a sua própria cópia.

Observação 3: caso você faça a parte de codificação em dupla ou em trio, insira no início do código um comentário com o nome de todos os autores. Os nomes devem estar nas primeiras linhas, antes mesmo dos #includes. Caso você não faça isso, o Run Codes acusará que você plagiou de sua dupla (ou trio).

Observação 4: para auxiliar na codificação, consulte o arquivo [RedBlackBST.java](#), criado por Sedgewick.

Observação 5: as operações de pesquisa e de percursos são idênticas às das árvores binárias de busca, como apresentado em Algoritmos e Estruturas de Dados I, e também às da Aula 06.