



UNIVERSIDADE FEDERAL RURAL DE PERNAMBUCO
Rua Dom Manoel de Medeiros, s/n – Dois Irmãos 52171-900 Recife-PE
Fone: 81 3302 1000 www.dc.ufrpe.br

DISCIPLINA: Algoritmos e Estruturas de Dados	CÓDIGO: 06214
DEPARTAMENTO: Computação	ÁREA: Informática
CURSO: Licenciatura em Computação	
PROFESSOR RESPONSÁVEL: Luciano Demétrio Santos Pacífico	
DATA MÁXIMA DE ENTREGA: 10-10-2020	

Regras da Lista de Exercícios 07

1. **Não é permitido o uso de Estruturas de Dados prontas de Linguagens de Programação.** O aluno deve implementar suas próprias Estruturas de Dados. Na Linguagem de Programação **C**, deve-se usar **structs**. Nas demais Linguagens de Programação permitidas (vide **Regras da Disciplina**), deve-se usar **classes**.
2. **Não é permitido o uso de Algoritmos e comandos otimizados prontos de Linguagens de Programação.** Todos os algoritmos solicitados devem ser implementados pelos alunos como **procedimentos** (funções, métodos, etc.).
3. As questões que solicitam **escrita de código** devem ser resolvidas **apenas através dos recursos oferecidos pela pseudolingagem definida para a disciplina, e dos recursos equivalentes em Linguagens de Programação reais**, sendo eles: variáveis, constantes e tipos primitivos, expressões, estruturas condicionais, estruturas de repetição, sub-rotinas, estruturas de dados homogêneas (**Arrays**) e estruturas de dados heterogêneas (registros – classes e structs).
4. Para a Lista de Exercícios 07, **todas as Estruturas de Dados devem ser Encadeadas (Dinâmicas)**, não sendo permitida a alocação de memória de forma **Sequencial (Estática)**. **As Estruturas de Dados devem ser implementadas pelos alunos**, não sendo permitido o uso de Estruturas de Dados Dinâmicas implementadas em Linguagens de Programação, como os **Vectors** e **Lists** da Linguagem de Programação **Java**, ou **Lists** da Linguagem **Python**, por exemplo.
5. **Regra de Ouro:** Todos os alunos envolvidos em **cópias** terão suas notas **ANULADAS** nas referidas questões.
6. Apenas o código “.c”, “.cpp”, “.java”, “.py”, etc. deve ser enviado ao professor para cada questão. Deve-se enviar **um único arquivo resposta por questão**, que conterá todas as classes/estruturas e procedimentos necessários para a solução da questão. Todos os arquivos devem ser enviados **em uma única pasta, “zipados”**.
7. O arquivo de resposta com o código para cada questão deve ser nomeado na forma “L#Q%.c”, “L#Q%.java”, etc., onde “#” refere-se ao número da lista e “%” refere-se ao número da questão (ex.: L1Q2.c, para o arquivo de resposta da segunda questão da primeira lista, usando a Linguagem C).
8. A resposta do Lista de Exercício 07 deve ser submetida **unicamente através da tarefa criada no Google Classroom para este propósito**.

Lista de Exercícios 07 – Árvores AVL

1. Implemente, em uma **Linguagem de Programação**, as **Estruturas de Dados** e algoritmos necessários, para uma **árvore AVL** binária dinâmica, que após a **inserção** de n nós **mantenha-se balanceada**. Os **Conjuntos de Entradas** para essa questão serão fornecidos através de **Arquivos**. Deve-se gerar um **arquivo de saída**, de acordo com o padrão especificado em detalhes na sequência. Recomenda-se o

uso da ferramenta **Notepad++** para a correta visualização do formato dos arquivos de entrada e do arquivo exemplo de saída. (10.0 pontos)

Entrada: O arquivo de entrada consistirá de vários conjuntos de dados. Cada linha do arquivo conterá n elementos, representando valores de chaves dos nós que formarão a estrutura de dados. Cada valor deverá ser inserido na árvore de acordo com a sua ordem no conjunto de dados. Todos os elementos em cada conjunto de dados terão valores de chave diferentes.

Saída: Para cada conjunto de dados, o arquivo de saída deve conter os resultados de acordo com o que se pede abaixo.

Para cada conjunto de dados, a cada elemento inserido deve-se exibir uma linha contendo uma das mensagens especificadas na tabela abaixo, de acordo com o resultado da inserção.

Mensagem	Causa
“arvore ja balanceada.”	Quando, após a inserção do novo nó, a árvore permanece AVL.
“rotacao direita.”	Quando, após a inserção do novo nó, é necessária a realização de uma rotação direita para que a árvore vire uma AVL novamente.
“rotacao esquerda.”	Quando, após a inserção do novo nó, é necessária a realização de uma rotação esquerda para que a árvore vire uma AVL novamente.
“rotacao direita dupla.”	Quando, após a inserção do novo nó, é necessária a realização de uma rotação direita dupla para que a árvore vire uma AVL novamente.
“rotacao esquerda dupla.”	Quando, após a inserção do novo nó, é necessária a realização de uma rotação esquerda para que a árvore vire uma AVL novamente.

Caso seja necessária a realização de alguma rotação, antes da impressão de uma das mensagens acima, deve-se imprimir a mensagem “no responsavel: #” (onde “#” representa a chave do nó responsável pela infração), além de cada elemento da árvore, em ordem, com o valor de seu fator de balanceamento entre parênteses. Os elementos devem ser impressos separados por um espaço em branco, e a linha com os elementos deve terminar com um espaço em branco.

Na linha seguinte à impressão da mensagem, deve-se imprimir cada elemento da árvore (nesse ponto, os elementos já deverão estar com os valores finais corrigidos, quando alguma correção for necessária), em ordem, com o valor de seu fator de balanceamento entre parênteses. Os elementos devem ser impressos separados por um espaço em branco e cada linha com os elementos deve terminar com um espaço em branco. A terceira linha após uma inserção deve conter o valor da altura total da árvore.

Entre cada conjunto de dados, uma linha em branco deve ser impressa. As últimas duas linhas do arquivo de saída devem estar em branco.

Para a resolução da questão, considerar que o fator de balanceamento de um nó é dado pela fórmula abaixo:

$$fb(v) = h(v.dir) - h(v.esq)$$

onde v é o nó avaliado, $v.dir$ é a subárvore direita do nó v e $v.esq$ é a subárvore esquerda de v .

Verifique os arquivos de entrada e saída exemplos (L7Q1_in.txt e L7Q1_out.txt, respectivamente).