Instituto Federal de Pernambuco - Campus Belo Jardim
Curso de Bacharelado e Engenharia de Software
Algoritmos e Entrutura de Dados - Prof. Lucas Sampaio Leite

Lista de Exercícios 03

Instruções:

- Esta lista deve ser respondida individualmente e contempla os conteúdos abordados nas aulas 12 a 15.
- As respostas devem ser submetidas por meio do formulário disponibilizado no site da disciplina, utilizando obrigatoriamente os arquivos template fornecidos.
- Não serão aceitas submissões após o encerramento do prazo estabelecido.
- Esta lista comporá parte da nota da Avaliação 2.

Problemas

1. Pesquise uma biblioteca da linguagem de programação de sua preferência que permita medir o tempo de execução de um programa. Utilize essa biblioteca para registrar o instante de início da execução e o instante de término, calculando e exibindo o tempo total de execução. Pesquise também uma biblioteca para geração de gráficos, que será utilizada para apresentar os resultados obtidos.

a) Implemente os algoritmos de busca linear e busca binária na linguagem escolhida. Realize experimentos utilizando vetores ordenados de diferentes tamanhos (pequenas, médias e grandes entradas), registrando o tempo de execução de cada algoritmo. Apresente gráficos com os resultados obtidos e faça uma análise comparativa do desempenho observado, relacionando-o com a complexidade assintótica estudada em sala de aula. Submeta como resposta o código, os gráficos, os resultados obtidos e a justificativa da análise realizada.

b) Implemente os algoritmos de ordenação Bubble Sort, Selection Sort, Insertion Sort, Merge Sort e Quick Sort. Varie o tamanho das entradas (pequenas, médias e grandes) e analise empiricamente o tempo de execução de cada algoritmo. Apresente gráficos comparando os resultados obtidos e discuta se o comportamento observado está de acordo com a

complexidade assintótica de cada algoritmo. Submeta o código, os gráficos, os resultados obtidos e a justificativa da análise realizada.

2. Implemente uma BST (Binary Search Tree) em C contendo as seguintes funcionalidades:

a) Inserção: deve ser capaz de inserir um dado elemento. Não há retorno.

b) Busca por elemento: recebe um elemento e retorna o valor do elemento buscado, caso esteja presente na árvore, ou null, caso contrário.

c) Remoção: deve remover um dado elemento, caso ele esteja presente na árvore. Não há retorno. Se o nó a ser removido possuir grau 2, sua implementação deve adotar a estratégia de substituição pelo nó determinado a partir da subárvore de maior altura. Caso as alturas das duas subárvores sejam iguais, substitua pelo nó determinado a partir da subárvore da esquerda.

d) Altura da árvore: deve retornar a altura da árvore ou -1, caso a árvore seja vazia.

e) Profundidade do nó: recebe um valor e deve retornar a profundidade do nó que o contém ou -1, caso o valor não exista na árvore.

f) Limpar a árvore: deve remover todos os elementos da árvore.

Obs.: Para facilitar a implementação, os nós da árvore devem armazenar valores inteiros.

3. Implemente uma árvore AVL que armazene um tipo Estudante, que possui uma chave, nome e CPF. Sua AVL deve prover as seguintes funcionalidades básicas:

a) Inserção: Deve ser capaz de inserir um dado elemento e manter-se balanceada. Não há retorno.

b) Busca por elemento: Recebe um elemento e retorna o valor do elemento buscado caso esteja presente na árvore, ou null, caso contrário.

c) Remoção: Deve remover um dado elemento, caso ele esteja presente na árvore, e manter-se balanceada. Não há retorno.

INSTITUTO FEDERAL
Pernambuco

Questões Beecrowd

1244 - Ordenação por Tamanho
1382 - Elementar, meu Caro Watson!
1191 - Recuperação da Árvore
1195 - Árvore Binária de Busca