Universidade Federal do Piauí – UFPI
Campus Senador Helvídio Nunes de Barros – CSHNB
Curso de Sistemas de Informação Bloco: IV
Disciplina: Estruturas de Dados II
Professora: Juliana Oliveira de Carvalho
Acadêmico:.....Matrícula:.....

TRABALHO DE ESTRUTURAS DE DADOS II PARA A PRIMEIRA AVALIAÇÃO

1) **(1,0 ponto)** Faça um programa em C que gere 1000 números aleatórios, insira-os em uma árvore binária de busca, incialmente vazia. Depois imprima o nível da folha de maior profundidade e o nível da folha de menor profundidade. Repita o processo 30 vezes. Depois mostre quantas das 30 vezes a diferença entre a profundidade máxima e mínima foram de 0, 1, 2 , 3 e assim por diante. Também contabilize o tempo gasto para inserir todos os elementos na árvore. E depois em cada uma das inserções faça experimento de verificar o tempo de busca de elementos, use sempre os mesmo elementos em todos os experimentos.

2) **(1,0 ponto Extra)** Repita todo o processo para uma árvore AVL e depois para uma árvore vermelho-reta. Compare os resultados obtidos.

3) **(2,5 pontos)** Cada unidade em um livro-texto inglês contém um vocabulário Inglês-Português de palavras que são usadas pela primeira vez em uma unidade em particular. Faça um programa que converta um conjunto de tais vocabulários armazenados em um arquivo Inglês em um conjunto de vocabulários Português-Inglês. Para isso considere que:
   (a) Os nomes das unidades são precedidos por um símbolo de porcentagem.
   (b) Há somente uma entrada por linha.
   (c) Uma palavra em Inglês é separada por dois pontos de sua(s) equivalente(s) portuguesa(s); se há mais de uma equivalente, elas são separadas por vírgula.

   Para produzir as palavras em ordem alfabética, crie uma árvore binária de busca para cada unidade que contenha as palavras em português, e listas ligadas das equivalentes inglesas. Assegure-se de que exista apenas um nó para cada palavra portuguesa na árvore. Além disso, permita o usuário informar uma unidade e então imprima todas as palavras da unidade em português seguida das equivalentes em inglês. Depois. imprima todas as palavras em inglês equivalentes a uma palavra em português dada, pesquisando para isso em todas as árvores das unidades cadastradas. Por fim permita o usuário remover uma determinada palavra em português de uma unidade, e consequentemente as palavras em português equivalentes a ela.

4) **(1,0 ponto)** Repita o exercício 3 para uma árvore 2-3. E depois compare o tempo de inserção e de busca na árvore do exercício 3 e do exercício 4.

5) **(2,5 pontos)** Suponha que uma memória seja dividida em blocos lógicos de 1Mbyte, e que o primeiro bloco é o bloco 0, suponha também que o gerenciador de memória de um Sistema Operacional mantenha uma árvore 2-3-4 (variação da árvore B) com nós para blocos livres e ocupados da memória. Cada nó da árvore 2-3-4 contém os seguintes campos: O- para ocupado ou L- para livre, o número do bloco inicial, o número do bloco final, endereço inicial (correspondente ao endereço inicial do primeiro bloco do nó) e endereço final (correspondente ao endereço final do último bloco do nó). Os nós da árvore será organizados de forma que um nó ocupado está entre 2 nós livres e vice-versa.

   (a) Faça um programa em C que cadastra os nós da árvore, onde o usuário deve informar se o primeiro nó é livre ou ocupado, o endereço inicial e final do nó. Os demais nós serão contabilizados pelo sistema se são livres ou ocupados e o usuário deve apenas informar o endereço inicial e final de cada um. O cadastro termina quando o usuário informar como endereço final de um nó o último endereço da memória.

   (b) Faça uma função que o usuário informe a quantidade de nós que ele precisa alocar e retorne as informações do nó que atenda as necessidades do usuário e então modifique a situação do referido nó de Livre para Ocupado.
      (i) Lembre-se que a árvore deve manter nós intercalados de acordo com a situação do nó, ou seja, se a situação de um nó muda então os nós adjacentes a ele deve ser concatenados. Consequentemente os nós da árvore serão modificados.

(ii) Lembre-se também se o nó escolhido possui uma quantidade maior de blocos do que o solicitado pelo usuário os nós árvore devem ser atualizados de forma que mantenha blocos adjacentes livres ou ocupados em um mesmo nó.

(c) Faça uma função em que o usuário informe blocos que devem ser liberados.
   (i) Lembre-se que a árvore deve manter nós intercalados de acordo com a situação do nó, ou seja, se a situação de um nó muda então os nós adjacentes a ele deve ser concatenados. Consequentemente os nós da árvore serão modificados.
   (ii) Lembre-se também se o nó escolhido possui uma quantidade maior de blocos do que o solicitado pelo usuário os nós árvore devem ser atualizados de forma que mantenha blocos adjacentes livres ou ocupados em um mesmo nó.

6) **(1,0 ponto)** Em relação ao exercício anterior verifique o tempo médio para encontrar um nó na árvore, bem como para alterar a árvore quando um nó é modificado.
   (a) Lembre-se de para cada caso executar 30 vezes para fazer a média.
   (b) Lembre-se de executar casos diferentes, por exemplo: casos em que o nós não é particionado e casos em que o nó é particionado; casos em que o nó buscado está em um nível maior do que outro.
   (c) Por fim, faça uma comparação e análise dos resultados.

7) **(1,5 ponto)** Repita o exercício 5 e 6 usando agora uma árvore AVL.
8) **(0,5 ponto)** Faça uma comparação entre resultados obtidos dos exercícios usando Árvore 2-3-4 e Árvore AVL.

Equipe: os programas podem ser feitos em dupla, mas os relatórios são individuais. Se os programas forem feitos em dupla, a dupla deve ser identificada no envio do código.
Data de Entrega: data primeira prova escrita
Entregar: Código Fonte, Relatório(Conforme Modelo em PDF)
Forma de Entrega: pelo SIGAA, caso tenha algum problema enviar por e-mail(julianaoc@ufpi.edu.br).
Entrevista Individual: agendar horário com a Professora.