Universidade Federal do Ceará
Campus Russas

Estruturas de Dados Avançadas
Trabalho 1

As implementações devem ser apresentadas em Linguagem C e os algoritmos devem seguir os que foram vistos nas aulas.

1. Tabela de Dispersão (Hash)

Implementar as funções de dispersão (funções hash) considerando os métodos:

(a) Método da divisão. Considere que a função recebe dois parâmetros (o valor da chave e o tamanho da tabela)

(b) Método da dobra

(i) considere que as chaves são números binários e que o tamanho da tabela é potência de 2 (isto é, $2^1 = 2$, $2^2 = 4$, $2^3 = 8$, $2^4 = 16$,...). Faça uma função de dispersão para cada uma das operações: 'e', 'ou' e 'ou exclusivo'.

(ii) agora considere que as chaves são números decimais e que o tamanho da tabela é uma potência de 10 (isto é, 10, 100, 1000, 10000,...). Faça uma função de dispersão para operação '+' (lembre-se de não considerar o vai 1).

(c) Método da multiplicação. Considere que as chaves são números decimais e que o tamanho da tabela é uma potência de 10 (isto é, 10, 100, 1000, 10000,...)

(d) Método da análise de dígitos

(i) considere que as n chaves são números decimais, que o tamanho da tabela é uma potência de 10 (isto é, 10, 100, 1000, 10000,...) e que a função recebe 4 parâmetros: o valor da chave, o números de dígitos que serão escolhidos da chave, um vetor com todas as chaves para calcular a distribuição e o tamanho do vetor das chaves.

(ii) faça uma função de dispersão para cada um dos desvios de distribuição:

$$\sum_{i=0}^{9} \left(n_i - \frac{n}{10}\right)^2 \text{ e } \sum_{i=0}^{9} \left|n_i - \frac{n}{10}\right|$$

2. Tratamento de colisões

Implementar os tratamentos de colisão pelo método exterior por **listas encadeadas** e por por **árvores AVL**, isto é, num método temos uma lista encadeada para cada compartimento da tabela, enquanto o outro método temos uma AVL para cada compartimento da tabela.

3. Relatório de Comparação dos Métodos

Crie 5 tabelas (vetores) de tamanho 200.000:

- 1 tabela para o Método da divisão.
- 1 tabela para o Método da dobra (considerando as chaves decimais)

- 1 tabela para o Método da multiplicação
- 2 tabelas para o Método da análise de dígitos (uma tabela para cada desvios de distribuição)

Crie um vetor de tamanho n para armazenar as chaves que deverão ser escolhidas randomicamente entre 0 e 2.000.000.000.

Depois percorra o vetor das n chaves inserindo cada chave em cada uma das 5 tabelas utilizando o método específico de cada tabela (divisão, dobra,...). Isto é, em cada uma das 5 cinco tabelas deverão ser inseridas as n chaves.

Para efeito de comparação dos métodos de tratamento de colisão, contar os tempos dessas duas abordagens pela buscar de 1.000.000 chaves aleatóreas.

O processo do relatório descrito acima deverá ser refeito 5 vezes considerando os valores para n:

- $n = 50.000$
- $n = 100.000$
- $n = 150.000$
- $n = 200.000$
- $n = 250.000$

O relatório deverá apresentar um gráfico (quantidade de chaves n) $\times$ (nº de colisões) em que é mostrado as 5 curvas (uma curva para cada um das tabelas) com os 5 valores n pedidos acima.

Isto é, pelo gráfico podemos ver quantas colisões em cada um dos 5 métodos de função de dispersão ocorreram para $n = 50.000$ chaves, para $n = 100.000$ chaves,...

4. AVL - implementações

1. Implementar as funções inserir e remover para AVL. Essas implementações devem seguir os algoritmos vistos em aula.
2. Implementar uma função que verifica se uma árvore é AVL fazendo o cálculo das alturas das subárvores de cada nó e verificando se o campo 'bal' de cada nó está realmente correto.
3. Implementar uma função que conta a quantidades de nós de uma AVL.

5. AVL - Testes

1. Deve-se criar 1.000 AVL's
2. Em cada AVL deve-se inserir aleatoriamente 10.000 nós onde a chave de cada nó está entre 0 e 100.000 (verificar se a AVL possui os 10.000 pelo algoritmo de contagem de nós)
3. Após todas as inserções verificar se a árvore é AVL pelo algoritmo de verificação.
4. Remover 1.000 nós (verificar se a AVL possui os 9.000 nós pelo algoritmo de contagem de nós)
5. Após todas as remoções verificar se a árvore é AVL pelo algoritmo de verificação.

O processo anterior pode ser realizado com uma AVL por vez, isto é, não é necessário ter 1.000 AVL's ao mesmo tempo em sua memória, mas apenas 1 por vez.

6. Apresentação

A apresentação será por arguição e será realizada nos dias informados no plano de ensino.

7. Entrega dos Códigos e do Relatório

Os códigos e o relatório devem ser entregues pelo SIGAA até o término da tarefa.

OBS.: Caso ocorra problemas com memória, considerar valores mais baixos na sua implementação ou uso de memória secundária (se isso ocorrer deve ser informado no relatório também).