

BCC202 – Estruturas de Dados I (2013-02)

Departamento de Computação - Universidade Federal de Ouro Preto - MG

Professor: **Reinaldo Fortes** (www.decom.ufop.br/reinaldo)

Estagiária docente: **Josiane Rezende**

Monitores: **Bruno H. M. dos Santos**



Trabalho Prático 03 (TP03)

- **Data de entrega: 07/02/2013 até 23:55. O que vale é o horário do Moodle, e não do seu, ou do meu, relógio!!!**
- **Decréscimo por atraso de até:** 12h = **30%**; 24h = **40%**; 36h = **60%**; 48h = **70%**; **Acima de 48h = 100%**.
- O padrão de entrada e saída deve ser respeitado **exatamente** como determinado no enunciado.
- **Parte da correção é automática, não respeitar as instruções enunciadas pode acarretar em perda de pontos.**
- **Bom trabalho!**

1 Objetivos

Este trabalho prático tem como objetivo principal fundamentar as principais estruturas de dados e os algoritmos de pesquisa apresentados na disciplina.

2 Descrição do Problema

Utilizando-se da TAD `TContato` definida na *Prática 04*, implemente as seguintes TADs de pesquisa:

1. Pesquisa sequencial.
2. Pesquisa binária.
3. Árvore binária de pesquisa.
4. Árvore AVL.
5. Tabela hash (utilizando listas encadeadas para tratar colisões).
6. Tabela hash (utilizando a estratégia de endereçamento aberto para tratar colisões).

As TADs de pesquisa terão como parte de seus dados um conjunto de contatos que deverá ser representado internamente pela estrutura de dados que melhor convier para cada algoritmo de pesquisa (*array* ou *lista encadeada*, por exemplo).

Cada TAD de pesquisa deverá possuir pelo menos duas operações:

- **Construir:** *contrói* a estrutura interna para armazenamento dos contatos. Recebe como argumento de entrada um vetor de contatos.
- **Pesquisa:** realiza a pesquisa por um determinado elemento utilizando o algoritmo correspondente. Recebe como entrada a chave do elemento procurado e retorna se a pesquisa encontrou o elemento procurado e o número de *comparações feitas*.

2.1 Entrada e Saída

A **entrada** inicia com o número de casos de teste (**C**) na primeira linha. Em seguida, separados por uma linha em branco, a definição de cada caso de teste. Um caso de teste é definido inicialmente pelo número de contatos a serem considerados (**N**) e o número de pesquisas a serem feitas (**P**), ambos na primeira linha de definição do caso de teste. A seguir, serão definidas **N** linhas contendo as informações de contato: nome, aniversário e telefone (conforme padrão

adotado na *Prática 04*). Logo após a definição dos contatos, seguirão **P** linhas contendo os números de telefone a serem pesquisados. Não deverão existir chaves repetidas na entrada.

A **saída** consistirá do número de telefone procurado, seguido de seis linhas com o resultado dos métodos de pesquisa, na ordem em que eles foram definidos anteriormente, contendo as seguintes informações: **1**, caso o método tenha encontrado o elemento, **0** caso contrário, seguido pelo número de comparações feitas.

A tabela a seguir apresenta um exemplo do padrão de entrada e saída. Os números de comparações feitas foram definidos ao acaso, portanto, não condizem com a realidade.

Entrada	Saída
2	31-8888-3333
	1 100
5 2	1 101
Fulano1 0128 31-8888-1111	1 102
Fulano2 0225 31-8888-2222	1 103
Fulano3 0323 31-8888-3333	1 104
Fulano4 0420 31-8888-4444	1 105
Fulano5 0518 31-8888-5555	
31-8888-3333	31-8888-6666
31-8888-6666	0 10
	0 20
4 2	0 30
Fulano4 0420 31-8888-4444	0 40
Fulano5 0128 31-8888-5555	0 50
Fulano6 0225 31-8888-6666	0 60
Fulano7 0323 31-8888-7777	
31-8888-5555	31-8888-5555
31-8888-8888	1 110
	1 121
	1 132
	1 143
	1 154
	1 165
	31-8888-8888
	0 15
	0 26
	0 37
	0 48
	0 59
	0 60

3 Imposições e comentários gerais

- Seu programa não pode ter “*memory leaks*”, ou seja, toda memória alocada deve ser liberada pelo seu código.
- Clareza, identificação e comentários no código também serão avaliados. Por isso, escolha cuidadosamente o nome das variáveis e torne o código o mais legível possível.
- Trabalhos copiados (e FONTE) terão nota zero, além de os alunos envolvidos no plágio perderem toda a nota atribuída a participação e pontos extras, entre outros (...). **Isto vale para qualquer entregável** (seção 4).
- Caso seja necessário, alunos poderão ser convocados para entrevista.

4 Entregáveis

Deverão ser entregues para avaliação do trabalho:

- **Código fonte:** código fonte do programa em C.
- **Arquivos de entrada e saída:** arquivos de entrada e saída utilizados para testar seu programa. Não será considerada entrada igual ao exemplo dado neste enunciado, procure elaborar um conjunto de testes abrangente, envolvendo vários contextos distintos (número de elementos, posição do elemento procurado e procura por elementos inexistentes).
- **Documentação:** a documentação deve ser entregue em um único arquivo PDF, e conter:
 1. **Implementação:** descrição da implementação do programa. Não faça “print screens” de telas e não inclua o código fonte. Ao contrário, procure resumir ao máximo a documentação, fazendo referência ao que julgar mais relevante. É importante, no entanto, que seja descrito o funcionamento das principais funções e procedimentos utilizados, bem como decisões tomadas relativas aos casos e detalhes de especificação que porventura estejam omissos no enunciado.
 2. **Análise comparativa dos métodos:** descreva e discuta os resultados dos testes, fazendo uma análise comparativa dos métodos de pesquisa em relação ao número de comparações.
 3. **Impressões gerais:** descreva o seu processo de implementação deste trabalho. Aponte coisas que gostou bem como aquelas que o desagradou. Avalie o que o motivou, conhecimentos que adquiriu, entre outros.
 4. **Conclusão:** conclusões e comentários gerais sobre o trabalho.

4.1 Como fazer a entrega

Verifique se seu programa compila e executa na linha de comando (utilizando o compilador GCC) antes de efetuar a entrega. Quando o resultado for correto, entregue via Moodle dois arquivos .ZIP e um .PDF com seu nome e sobrenome. Exemplo:

- **PrimeiroNome-UltimoNome.zip:** este arquivo .ZIP deve conter apenas os arquivos .c e .h, utilizados na implementação, e os arquivos .TXT de entrada e saída, utilizados para testar seu programa.
- **PrimeiroNome-UltimoNome.pdf:** este arquivo .PDF deve conter a documentação do trabalho.

Serão criadas duas tarefas no moodle:

1. Entrega dos arquivos de código fonte e arquivos de entrada e saída.
2. Documentação (arquivo .PDF).