

## SCC0203 - Algoritmos e Estruturas de Dados II – Profa. Cristina Oliveira

### Terceiro Trabalho Prático – Árvores B

Prazo máximo: 23:55h de 05/07/2012

Este terceiro trabalho prático é de realização opcional. A nota do mesmo pode substituir a nota de algum dos outros dois trabalhos realizados. Este trabalho deve ser feito **individualmente**.

Para este trabalho, não será usado o Online Judge. Deve ser entregue um arquivo compactado no Moodle, na pasta correspondente. O arquivo compactado devem ser nomeado com o número USP do aluno e conter a extensão “.zip” (ex: 1234.zip) . Cada arquivo compactado deve conter:

- **Um único arquivo de código fonte “.c”**, contendo a solução do problema. O arquivo deve ser denominado da seguinte maneira: Qx\_XXXX.c, onde: x é o número da questão. A seguir, deve-se colocar o número USP do autor. Ainda, dentro do código fonte deve-se ter, nas primeiras linhas, um comentário contendo o nome e o número USP do autor do trabalho.
- **Um relatório** de, no máximo, 2 páginas, explicando a teoria aplicada para a resolução do problema. Este relatório não deve conter detalhes de codificação do problema.

### Observações

- Deve-se fazer a implementação em linguagem **ANSI C**.
- O sistema bloqueia entregas após o prazo estipulado. Assim sendo, não será possível entregar o trabalho após este horário.
- Considere as entradas e saídas **ESTRITAMENTE** nos padrões descritos. Não imprima na tela quaisquer outras coisas que não as saídas esperadas pelo programa.
- Para fins de avaliação, apenas a última submissão de cada problema é considerada. Será analisado o código fonte, especialmente os TADs utilizados. **Os trabalhos também serão inspecionados quanto a plágio.**

**Bom trabalho!**

## Paulo Roberto e a Fórmula 1

Paulo Roberto é, definitivamente, um fanático por automobilismo. Fã de carteirinha dos esportes a motor e piloto amador de kart, Paulo Roberto adora colecionar qualquer coisa que diga respeito a esportes a motor, desde pôsteres de pilotos e miniaturas de carros de corrida, a até mesmo vídeos e dados sobre corridas.

A atual empreitada de Paulo Roberto é com a história da Fórmula 1. Ele está tentando tornar montar um grande arcabouço com dados históricos da Fórmula 1. Seu primeiro objetivo é disponibilizar um sistema que contenha todos os dados dos pilotos que já passaram pela história da Fórmula 1. Para isso, Paulo Roberto buscou em diversas fontes na internet e compilou uma grande base de dados com todos os pilotos que já passaram pela categoria, contendo uma série de informações de seu interesse.

A primeira tentativa de Paulo Roberto para esse sistema foi utilizar um sistema que fizesse uma busca sequencial no arquivo contendo os dados dos pilotos. Para isso, Paulo Roberto gravou os dados coletados em um único arquivo. Cada piloto era representado por um registro de tamanho fixo, contendo os seguintes campos:

Campo	Tamanho (bytes)
ID	4
Nome	29
País	15
Títulos mundiais	1
Corridas	3
Poles	2
Vitórias	2

Assim, cada registro de um piloto era composto por 56 bytes. Para completar o tamanho dos campos em que o conteúdo não atingia o tamanho desejado, Paulo Roberto colocou símbolos “#”, tantos quantos fossem necessários. Por exemplo, o registro correspondente ao piloto Ayrton Senna ficaria da seguinte maneira:

```
0059Ayrton Senna#####Brazil#####31616541
```

Indicando: ID = 0059; Nome = Ayrton Senna; País = Brazil; Títulos mundiais = 3; Corridas= 161; Poles = 65; Vitórias = 41;

Uma vez feito esse sistema, utilizando este arquivo de dados, Paulo Roberto disponibilizou na internet seu sistema de busca, logo fazendo grande sucesso. E Paulo Roberto sempre atualizava o sistema, colocando dados relativos a mais e mais pilotos que encontrava em suas pesquisas.

Entretanto, à medida em que a base de dados crescia, começaram a surgir, dos seus usuários, reclamações relativas à demora para pesquisa na base. Em um sistema como esse, que há interação com usuário, o tempo de resposta tem que ser o mais rápido possível. Foi aí que Paulo Roberto percebeu que a busca sequencial no arquivo não era mais satisfatória. Era, então, necessário o uso de um índice para acessar os dados.

Foi, então, que Paulo Roberto chegou até a você. Ouvindo dizer que você estava estudando **Árvores B** na disciplina de Alg II, Paulo Roberto resolveu pedir a sua ajuda. Ele deseja que você faça um sistema que construa um índice dos dados, facilitando a busca e manipulação dos dados.

Para isso, Paulo Roberto criou uma lista de características do sistema a ser feito:

1. O sistema deve considerar **Árvores B** de ordem a ser especificada na entrada. **Logo, o primeiro input do usuário seria a ordem da árvore B.**
2. A árvore deve ser construída em memória primária (há memória disponível para isso). Para essa construção, o usuário deve especificar o arquivo de entrada. **Assim, o segundo input do usuário seria o caminho para o arquivo contendo os dados dos pilotos.** Seu sistema deve inserir os registros na árvore na ordem em que estes estão disponíveis no arquivo de entrada. Este arquivo de entrada com os dados dos pilotos está disponível para download na Coteia, com o nome "**dados\_pilotos.txt**";
3. Cada registro do arquivo de dados deve ser indexado pelo nome do piloto. Assim, na árvore, um registro deve ser representado pelo nome do piloto e pelo RRN para sua entrada no arquivo de dados;
4. O sistema deve possibilitar a **busca e inserção de registros** no arquivo de dados. Dessa forma, o sistema, após ler o arquivo de entrada e montar a árvore B, deve aguardar por comandos. Os comandos desejados, bem como o resultado esperado de cada um deles, estão listados a seguir.
  - a. **BUSCA(XXX)**: Deve buscar, por meio da árvore B, um determinado registro no arquivo de dados. "XXX" é o nome do piloto procurado. Seu programa deve imprimir o caminho percorrido na busca, imprimindo as chaves contidas nos nós percorridos. Ao encontrar o piloto desejado no arquivo de dados, seu programa deve imprimir, na tela, todos os dados deste piloto.
  - b. **INSERE(XXX)**: Deve inserir o registro no arquivo de dados, bem como na árvore de índice.

- c. **FIM:** este comando indica o fim da execução do seu programa. Neste ponto, todas as estruturas devem ser desalocadas e o processo encerrado.

### Exemplo de utilização do sistema

Abaixo, listamos uma ordem de execução do programa. Elementos em verde não devem ser impressos, estando presentes apenas para fins explicativos. Para fins didáticos, vamos considerar que o arquivo de entrada contém, em ordem, os registros relativos aos pilotos: Ayrton Senna, Bruno Senna, Alain Prost e Riccardo Patrese.

Início da execução do programa

Usuário fornece ordem da árvore  
3

Usuário fornece caminho para arquivo. Neste ponto, seu programa deve montar uma árvore inicial, com os dados constantes no arquivo de entrada e esperar pelos comandos do usuário, até que este entre com o comando FIM  
dados\_pilotos.txt

Usuário faz busca  
BUSCA(Riccardo Patrese)

Nós percorridos:  
Ayrton Senna  
Bruno Senna, Riccardo Patrese

Dados do piloto procurado:  
ID = 0694  
Nome = Riccardo Patrese  
País = Italy  
Títulos mundiais = 0  
Corridas = 256  
Poles = 8  
Vitórias = 6

Usuário insere novos registros. Estes registros devem ser inseridos no arquivo de dados e na árvore.

INSERE(0564Michael Schumacher#####Germany#####72936891)  
INSERE(0728Rubens Barrichello#####Brazil#####03221411)  
INSERE(0249Felipe Massa#####Brazil#####01581511)

Usuário faz novas buscas  
BUSCA(Ayrton Senna)

Nós percorridos:  
Ayrton Senna, Michael Schumacher

Dados do piloto procurado:  
ID = 0059  
Nome = Ayrton Senna  
País = Brazil  
Títulos mundiais = 3  
Corridas= 161  
Poles = 65  
Vitórias = 41

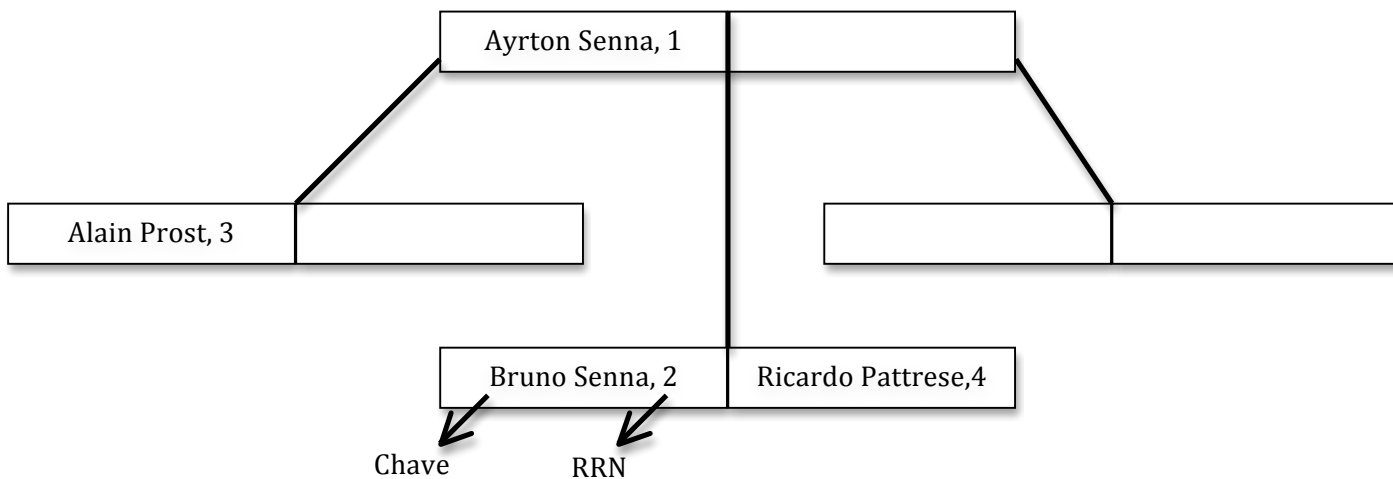
BUSCA(Felipe Massa)

Nós percorridos:  
Ayrton Senna, Michael Schumacher  
Bruno Senna, Felipe Massa

Dados do piloto procurado:  
ID = 0249  
Nome = Felipe Massa  
País = Brazil  
Títulos mundiais = 0  
Corridas = 158  
Poles = 15  
Vitórias = 51

Usuário encerra funcionamento do programa  
FIM

A árvore inicial, formada com os dados constantes no arquivo de dados, seria da forma:



Após a inserção de todos os registros, sua árvore vai ter a seguinte configuração:

