

Pontifícia Universidade Católica de Minas Gerais  
Curso de Ciência da Computação  
Disciplina: Algoritmos e Estruturas de Dados II  
Profs.: Felipe Domingos da Cunha, Max do Val Machado e  
Rodrigo Richard Gomes

# Trabalho Prático V

---

## Regras Básicas

- extends Trabalho Prático 04
- Fique atento ao Charset dos arquivos de entrada e saída.

### Observação:

**Não será necessário implementar a opção de remoção nas TADs abaixo.**

A National Basketball Association (em português: Associação Nacional de Basquetebol; abreviação oficial: NBA) é a principal liga de basquetebol profissional da América do Norte. Com 30 franquias sendo membros da mesma (29 nos Estados Unidos e 1 no Canadá), a NBA também é considerada a principal liga de basquete do mundo. É um membro ativo da USA Basketball (USAB), que é reconhecida pela FIBA (a Federação Internacional de Basquetebol) como a entidade máxima e organizadora do basquetebol nos Estados Unidos. A NBA é uma das 4 'major leagues' de esporte profissional na América do Norte. Os jogadores da NBA são os mais bem pagos esportistas do mundo, por salário médio anual.



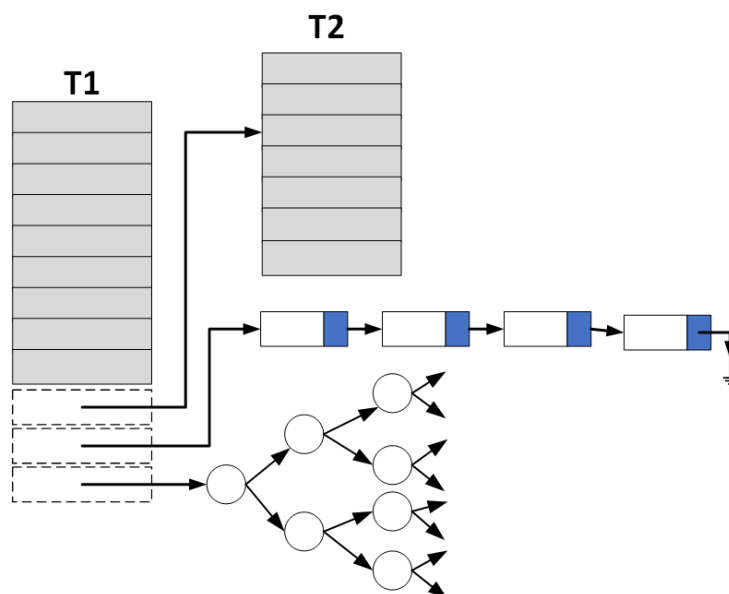
A liga foi fundada na cidade de Nova Iorque em 6 de Junho de 1946, como a Basketball Association of America (BAA). A liga adotou o nome de National Basketball Association em 1949 quando se fundiu com a rival National Basketball League (NBL). A liga tem diversos escritórios ao redor do mundo, além de vários dos próprios clubes fora da sede principal na Olympic Tower localizada na Quinta Avenida 645. Os estúdios da NBA Entertainment e da NBA TV são localizados em Secaucus, New Jersey.

O arquivo [players.csv](#) contém um conjunto de dados de jogadores da liga de basquete norte americana - NBA extraídos do site <https://www.kaggle.com/drgilermo/nba-players-stats>. Essa base contém registros de jogadores desde 1950. Um total de 67 temporadas da NBA. Este arquivo **sofreu algumas adaptações** para ser utilizado neste e nos próximos trabalhos práticos. Tal arquivo deve ser copiado para a pasta /tmp/. **Quando reiniciamos o Linux, ele normalmente apaga os arquivos existentes na pasta /tmp/.**

## Hash

Observação: **ATENÇÃO** para as estruturas hash que já estão implementadas no [Github!](#)

1. **Tabela Hash Direta com Reserva:** Refaça a primeira questão deste trabalho com Tabela Hash Direta com Reserva. A função de transformação será **altura mod tamTab** onde tamTab (tamanho da tabela) é 21. A área de reserva tem tamanho 9, fazendo com que o tamanho total da tabela seja igual a 30. A saída padrão será a posição de cada elemento procurado na tabela (na *hash* ou na área de reserva). Se o elemento procurado não estiver na tabela, escreva a palavra NÃO. Além disso, o nome do arquivo de log será matrícula\_hashReserva.txt.
2. **Tabela Hash Direta com Rehash:** Refaça a questão anterior com Tabela Hash Direta com *Rehash*. A primeira função de transformação será **altura mod tamTab** onde tamTab (tamanho da tabela) é 25 e a outra, **(altura + 1) mod tamTab**. O nome do arquivo de log será matrícula\_hashRehash.txt.
3. **Tabela Hash Indireta com Lista Simples em C:** Refaça a questão anterior com Tabela Hash Indireta com Lista Simples. A função de transformação será **altura mod tamTab** onde tamTab (tamanho da tabela) é 25. O nome do arquivo de log será matrícula\_hashIndireta.txt.
4. **Doidão em Java:** Seja a estrutura de dados Doidona abaixo para armazenar jogadores, crie a classe Doidona contendo os atributos necessários, o construtor e o método inserir. Você deve criar qualquer outra classe que seja necessária para que o processo de inserção aconteça sem erros.



Observações:

- (a) A figura apresenta duas tabelas: T1 e T2.
- (b) A tabela T1 de *hash* direta possui o tamanho **onze** e com área de reserva tamanho **três**.
- (c) A área de reserva de T1 possui T2, uma lista encadeada e um árvore binária de pesquisa de acordo com a figura.
- (d) A tabela T2 de *hash* direta com *rehash* e possui o tamanho **nove**.

A entrada dessa questão consiste em uma sequência de *ids* e a saída será a execução do **mostrar** em cada uma das estruturas. Lembrando que na árvore, vamos usar sempre o caminhamento *em-ordem*.