

# Pesquisa de Dados

Prof. Rafael Guterres Jeffman  
rafael.jeffman@gmail.com

# Conceitos Fundamentais

- Busca
  - Tentativa de encontrar informação numa massa de dados, a partir de um conjunto parcial de dados relacionados com a informação procurada.
- Registro
  - Uma coleção de dados agrupados de forma a representar alguma informação.
- Chave
  - Parte de um registro utilizada para a busca.

# Chaves de Procura

- Chave Interna
  - Parte do próprio registro.
- Chave Externa
  - Armazenada externamente ao registro.
- Chave Primária
  - Única para um conjunto de registros.
- Chave Secundária
  - Não é única para um conjunto de registros.

# Algoritmos de Busca

- Buscam por uma informação em um conjunto de dados a partir de um critério de busca (chave).
- O retorno de um algoritmo de busca é um conjunto de registros que atendem ao critério de busca.
- Pode não encontrar um registro, retornando dessa forma um conjunto vazio de resposta.

# Tipos de Busca

- Busca Simples
  - Retorna um conjunto de dados como resultado da busca.
- Busca e Inserção
  - Caso não encontre a informação, adiciona um novo registro com a chave de busca utilizada.
- Busca Interna
  - Todos os dados a serem pesquisados estão em memória.
- Busca Externa
  - Parte dos dados a serem pesquisados estão em armazenamento externo.

# Estruturas de Dados para Busca

- Várias estruturas de dados podem ser utilizadas para a execução do algoritmo de busca: arrays, listas encadeadas, árvores, tabelas *hash*
- Frequentemente, a técnica utilizada depende da organização dos dados.

# Busca Seqüencial

- Também conhecida como Busca Linear
- Aplica-se a dados estruturados de forma linear
- Cada registro é examinado a partir do início da lista e sua chave é comparada com a chave de busca.
- A busca continua até o elemento ser encontrado, ou a lista terminar.
- É a única forma de encontrar um registro em uma lista de dados desordenada.

# Busca Seqüênçial Otimizada

- Ao encontrar um registro, ele é movido para o início da lista, facilitando encontrar registros uma segunda vez.
- Ao encontrar um registro, ele é movido para uma posição anterior na lista, fazendo com que, gradativamente, registros mais acessados se posicionem no início da lista. (Transposição)



# Justificativa da Otimização

- A busca seqüencial é consideravelmente mais lenta que outras formas de busca.
- Existe uma possibilidade de registros acessados recentemente serem novamente acessados.
- Dessa forma, ao mover um elemento para o início da lista, as recuperações seguintes são mais rápidas.

# Busca em Listas Ordenadas

- Quando os dados estão organizados de forma ordenada, é possível procurar os registros de forma mais eficiente.
- Neste caso, uma otimização possível é que não é preciso procurar em toda a tabela quando o elemento não existe.

# Busca Seqüencial Indexada

- Utiliza uma Tabela de Índices para otimizar a busca.
- Dada uma coleção de itens com  $T$  elementos, e uma tabela de itens de  $N$  elementos, onde  $N < T$ , cada elemento  $x_N$ , com  $x = \{1, 2, 3, \dots, j\}$ , da coleção original terá uma entrada na tabela de índices.
- O ordenação das duas tabelas deve ser a mesma.

# Busca Seqüencial Indexada

- Diminui o número de comparações realizadas na busca.
- Pode ser utilizada em conjunto com listas encadeadas.
- Quando há vários registros com a mesma chave, o índice obtido não é necessariamente o primeiro registro com a chave.
- O tamanho da tabela de índices é grande demais ou pequeno demais com relação ao conjunto de dados.
- Podem ser utilizadas várias tabelas de índices encadeadas para resolver alguns dos problemas deste tipo de busca.

# Busca Binária

- Aplica-se apenas a conjuntos de registros ordenados.
- Requer acesso aleatório aos elementos da estrutura.

# Algoritmo de Busca Binária

- Avalie o elemento posicionado na metade da tabela de dados.
- Se a chave do elemento for maior que do elemento atual, o elemento procurado está na primeira metade da tabela.
- Se a chave do elemento for menor que do elemento atual, o elemento procurado esta na outra metade da tabela.