

• Ordenação (Ziviani)

→ algoritmos de ordenação bons exemplos de resolver problemas nos computadores

- conjunto amplo de algoritmos para resolver uma tarefa.

- dependendo da aplicação cada algoritmo considerado possui uma vantagem particular

↳ Ordenar

→ rearranjar um conjunto de objetos em ordem ascendente ou decrescente

→ objetivo: facilitar a recuperação posterior de itens do conjunto de dados ordenado

EX: dicionário
lista telefones
sumários

Computador - tabelas
- arquivos

- Grande parte dos métodos de ordenação é baseada em comparação das chaves

- Existem métodos que funcionam pelo princípio da distribuição

Ex: baralho

ordenar: a) $A < 2 < 3 \dots < 10 < J < Q < K$

e

b) $\clubsuit < \diamond < \heartsuit < \spadesuit$

- ordenar cartas por distribuição

1. distribuir cartas em 13 montes, um por caractere
2. colocar os montes na ordem citada (a)
3. redistribuir as cartas em 4 montes, um por naipe
4. reorganizar os montes

- Métodos conhecidos como:

- ordenação digital, radix sort, bucket sort
- não existem comparações entre as chaves
- problema: demanda por memória: $O(n)$

↳ Método de ordenação é dito **estável** se a ordem relativa dos itens com chaves iguais mantém-se inalterada pelo processo de ordenação

→ funcionários ordenados pelo salário
(mesmo salário \leftrightarrow ordem alfabética)

- alguns métodos de ordenação mais eficientes não são estáveis

- Métodos de Ordenação em dois grupos:

- ordenação interna: todos os dados cabem na memória principal

- ordenação externa: dados não cabem na memória principal, e por isso são armazenados em disco

diferença:

- interna: qualquer dado / registro pode ser imediatamente acessado

- externa: registros acessados sequencialmente ou em grandes blocos

* Ordenação Interna

- aspecto para escolha do algoritmo de ordenação é o tempo gasto para ordenar o arquivo

- A qtd de memória também. Uso econômico da memória é um requisito primordial

- Métodos Simples $O(n^2)$ comparações

- Métodos Eficientes $O(n \log n)$ comparações

- métodos simples são adequados para arquivos pequenos
- " eficientes para arquivos maiores

Simple: fácil de Entender

Eficientes: + complexos

- busca linear
 - busca ordenada
 - busca binária
- } Implementar as três funções

Busca linear: percorre até achar (ou não)

busca ordenada: percorre ordenadamente até achar (ou não)

busca binária: sempre divide ao meio (ordenado!)

Costos:

linear: $O(n)$

ordenada: $O(n)$

binária: $O(\log n)$

} Fazer plot