

Algoritmos e estrutura de dados I



Algoritmos de ordenação e busca

Introdução

Os algoritmos de busca e de ordenação compreendem um conjunto de algoritmos de elevada importância para a computação.

A ordenação consiste colocar uma lista/vetor em ordem (crescente/decrescente) numérica ou alfabética.

A busca consiste em localizar um elemento em uma lista.

Existem buscas que necessitam da ordenação.

Insertion sort

Insertion Sort, ou ordenação por inserção, é um algoritmo de ordenação

Dado uma lista, realiza a ordenação com um elemento de cada vez, ou seja, uma inserção por vez.

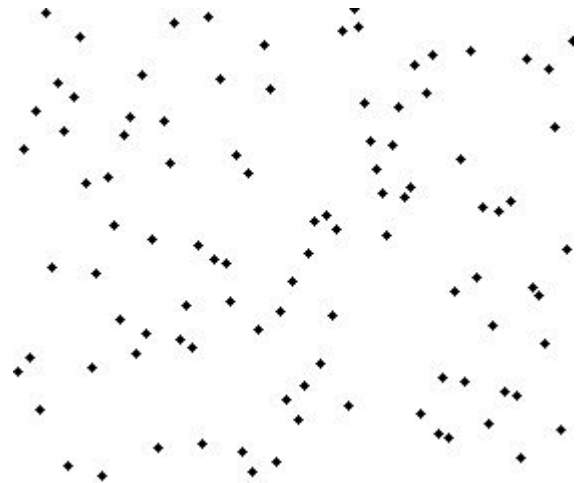
Podemos fazer uma comparação do Insertion Sort com o modo como algumas pessoas organizam um baralho num jogo de cartas.

Você está com as cartas na mão e elas estão ordenadas.
Você recebe uma nova carta e deve colocá-la na posição correta da sua mão de cartas, de forma a que as cartas obedeçam à ordenação.

Esta é a ideia por trás da ordenação por inserção.

Para compreender melhor a ideia do algoritmo, veja o vídeo abaixo:

[Dança insertion sort](#)



Bubble sort

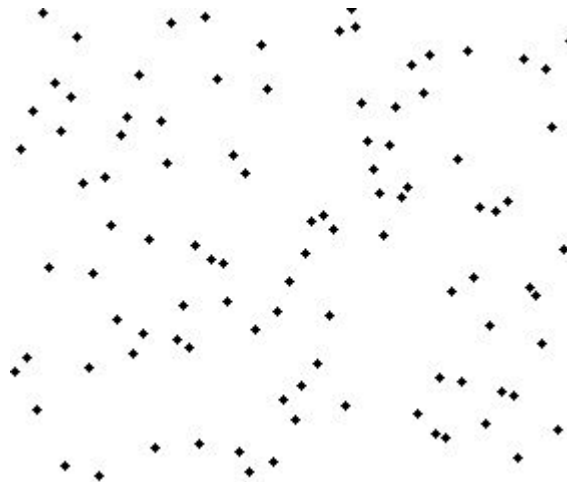
O bubble sort, ou ordenação por flutuação (literalmente "por bolha")

É um algoritmo de ordenação dos mais simples.

A ideia é percorrer o vetor diversas vezes, e a cada passagem fazer flutuar para o topo o maior elemento da sequência.

Para compreender melhor a ideia do algoritmo, veja o vídeo abaixo:

[Dança bubble sort](#)



Linear search

Na área de informática, ou Ciência da Computação, costuma-se usar o termo busca linear (ou busca sequencial) para expressar um tipo de pesquisa em vetores ou listas de modo sequencial.

Esse tipo de busca é realizado elemento por elemento.

Apesar de fácil compreensão, esse método não é o melhor para uma lista muito grande.

Para compreender melhor a ideia do algoritmo, veja o vídeo abaixo:

[Dança busca linear](#)

Binary search

É um algoritmo de busca em vetores que segue o paradigma de divisão e conquista.

Elas parte do pressuposto de que o vetor está ordenado

Realiza sucessivas divisões do espaço de busca comparando o elemento buscado (chave) com o elemento no meio do vetor. Se o elemento do meio do vetor for a chave, a busca termina com sucesso. Caso contrário, se o elemento do meio vier antes do elemento buscado, então a busca continua na metade posterior do vetor. E finalmente, se o elemento do meio vier depois da chave, a busca continua na metade anterior do vetor.

Para compreender melhor a ideia do algoritmo, veja o vídeo abaixo:

[Dança busca binária](#)