**ANTONIO MENEGHETTI FACULDADE**

**CURSO DE SISTEMAS DE INFORMAÇÃO**

**BRUNO VIERA**

**ALGORITMOS DE ORDENAÇÃO**

**RECANTO MAESTRO - RS**

**2016**

**RESUMO**

Os algoritmos de ordenação tem a finalidade de colocar elementos de uma sequência qualquer em uma certa ordem (completa ou parcial). Um exemplo comum de ordenação é a numérica.

**Palavras-chave:** Ordenação.Algoritmos.

**ABSTRACT**

Briefly, the sorting algorithms aims to place elements of a sequence any in a certain order ( complete or partial ) . A common example of ordering is the number .

**Keywords**: Ordination.Algorithms.

**1.INTRODUÇÃO**

Os algoritmos existem para conceder a resolução de um certo problema e cada algoritmo resolve uma instância desse problema. Um exemplo muito usado na computação é o de ordenação, que tem como finalidade ordenar uma lista que pode ser composta por palavras ou números, de acordo com a necessidade. Aqui serão explicados de forma clara e objetiva, o funcionamento dos seguintes algoritmos de ordenação: Shell; Cocktail; Inserção; Comb; Gnome.

**2.SHELL SORT**

O Shell sort é um algoritmo de ordenação sofisticado. A ideia desse algoritmo é dividir um grande vetor de dados em vetores menores, ordenando-os e fazendo isso novamente até ter um vetor ordenado para então trabalhar nele de forma prática e rápida.

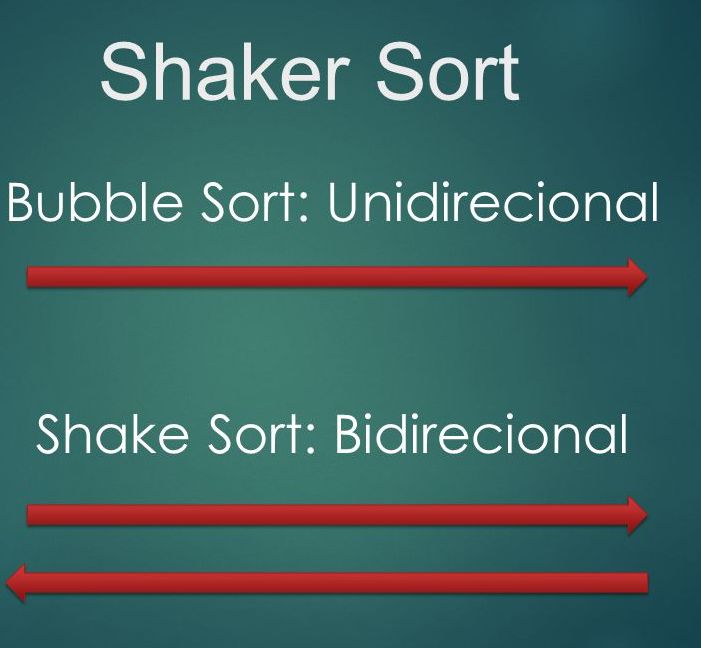
Funciona da seguinte forma: Utilizando o incremento de shell, o algoritmo compara elementos distantes em um vetor, em vez de comparar os adjacentes. A figura abaixo mostra como são selecionados os elementos do vetor utilizando a variável incremento de shell. Os valores selecionados são sempre de dois em dois pois h2=2



**3.COCKTAIL SORT**

O Cocktail sort é uma variação do Bubble sort que é tanto um algoritmo de ordenação estável quanto uma ordenação por comparação. A diferença se da pelo fato de ele ordenar em ambas as direções em cada passagem através da lista.

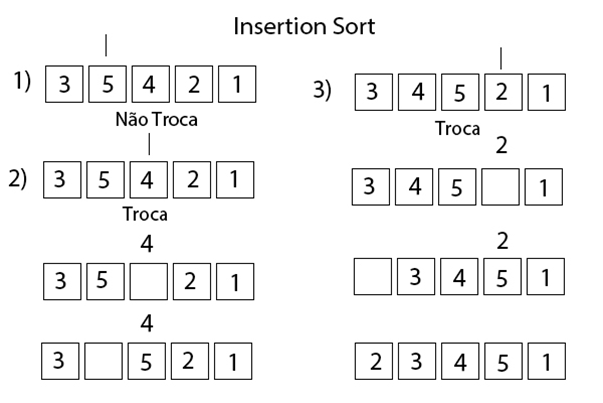
A maneira de trabalhar desse algoritmo é ir ordenando ao mesmo tempo pelos dois extremos do vetor, de maneira que depois da primeira interação, tanto o maior como o menor elemento estarão em suas posições finais. Desta forma, o número de comparações é reduzido .



**4.INSERTION SORT**

O Insertion sort mostra-se muito eficiente quando aplicado em pequenas listas. Ele percorre um vetor ordenando os elementos a esquerda a medida que avança.

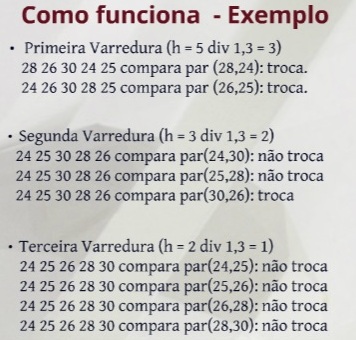
Funciona da mesma forma que as pessoas usam para ordenar as cartas em um jogo de baralho.



**5.COMB SORT**

O Comb sort é um algoritmo de ordenação relativamente simples e faz parte do grupo dos algoritmos de ordenação por troca.

Funciona da seguinte forma: O algoritmo repetidamente reordena diferentes pares de itens, separados por um salto, que é calculado a cada passagem. É semelhante ao Bubble sort, porém mais eficiente.

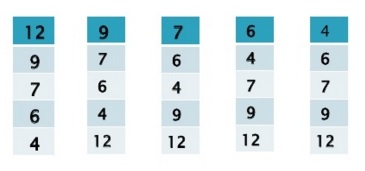


**6.GNOME SORT**

O Gnome sort é um algoritmo similar ao Insertion sort com a diferença que o Gnome leva um elemento para sua posição correta, com uma sequência grande de trocas assim como o Bubble sort. O algoritmo percorre o vetor comparando seus elementos dois a dois, assim que ele encontra um elemento que está na posição incorreta, ou seja, um número maior antes de um menor, ele troca a posição dos elementos, e volta com este elemento até que encontre o seu respectivo lugar.

Segue a explicação do exemplo abaixo: O número 12 é maior e está antes do número 9, então foi feita a troca de posições dos elementos e o algoritmo volta com o 12 fazendo a comparação dos elementos dois a dois e inserindo-o no lugar certo.

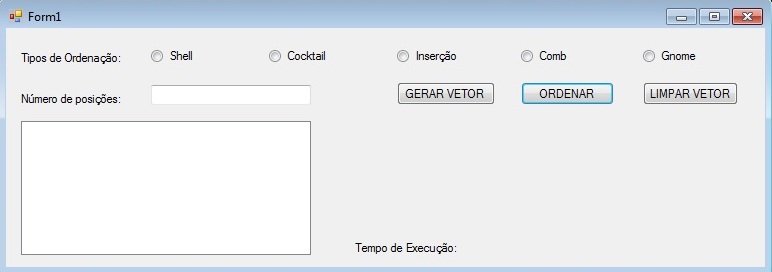
Após iniciar a comparação do 7 com o 6, trocamos as posições e o algoritmo volta com o 7 até inseri-lo na posição correta, desta maneira todos os elementos serão ordenados corretamente.



**7.TUTORIAL PARA EXECUÇÃO DO PROGRAMA**

A aplicação apresentada neste pequeno tutorial é de fácil entendimento por se tratar de uma aplicação de uso objetivo. Esta permite que o usuário possa gerar um vetor de inteiros aleatórios de no mínimo 100 (cem) posições. Ao lado de "Tipos de Ordenação" o usuário tem disponível cinco tipos de algoritmos de ordenação, são eles: Shell, Cocktail, Inserção, Comb e Gnome. Depois de escolher um tipo de ordenação, o usuário deve informar o número de posições que o vetor irá ter (lembrando que a aplicação não aceita números menores que 100) digitando dentro da caixa de texto localizada ao lado de onde se lê "Número de posições".

Feito isso, o usuário deve clicar no botão "GERAR VETOR", abaixo do número de posições encontra-se uma caixa de texto de tamanho maior que serve para mostrar o vetor requisitado pelo usuário. O vetor será mostrado de forma desordenada, mas para que os valores apareçam ordenados o usuário deve clicar no botão ORDENAR e após feita essa consulta, o botão LIMPAR VETOR limpa a tela da aplicação. E para cada execução, pode-se acompanhar o tempo que aparece ao lado de "Tempo de Execução"

****

**8. CONSIDERAÇÕES FINAIS**

A pesquisa desenvolvida aqui teve como objetivo mostrar alguns dos algoritmos de ordenação existentes e fazer uma breve explicação sobre seu funcionamento. Também mostra de forma clara, o funcionamento da aplicação feita em sala de aula nas últimas semanas.

**REFERÊNCIAS**

CADERNO GEEK. **Shell Sort.** Disponível em: https://cadernogeek.wordpress.com/tag/shell-sort/. Acesso em 21 de Abril de 2016.

DEVMEDIA. **Algoritmos de ordenação: análise e comparação.** Disponível em: http://www.devmedia.com.br/algoritmos-de-ordenacao-analise-e-comparacao/28261. Acesso em 21 de Abril de 2016.

PREZI. **Métodos de Ordenação.** Disponível em: https://prezi.com/4t2nvclqthvo/metodos-de-ordenacao/. Acesso em 21 de Abril de 2016.