
Estruturas de Dados

— Ordenação —

Baseado no material do Prof.
Yandre Gomes

Prof. Nilton Luiz Queiroz Jr.

Ordenação

- Entende-se por ordenação o processo de reorganizar os dados de um certo conjunto de objetos de acordo com um critério específico;

Ordenação

- Dado um conjunto de dados:

$$a_1, a_2, a_3, \dots, a_n$$

Ele é dito ordenado quando, dado uma função f de ordenação, a seguinte relação é satisfeita:

$$f(a_1) \leq f(a_2) \leq f(a_3) \leq \dots \leq f(a_n)$$

Ordenação

- Podemos dividir a ordenação em duas principais categorias:
 - Ordenação Interna: Usa-se somente a memória principal do computador, pois o volume dos dados não é tão grandes;
 - Ordenação Externa: Usa-se memória secundária devido ao grande volume de dados;

Ordenação

- Existem diversas maneiras de se implementar um método de ordenação, tanto interna quanto externa;
- Dentro dos métodos de ordenação existem algumas características relevantes:
- Para ordenação interna temos como tais características:
 - Tempo de execução;
 - Consumo de Memória;
 - Estabilidade;

Ordenação Interna

- Alguns métodos de ordenação consomem menos memória pois são facilmente feitos dentro do mesmo vetor;
 - Tais métodos podem ser chamados de *in situ*
- Dentre os métodos *in situ* existem 3 métodos de simples compreensão e implementação:
 - Ordenação por seleção (selection sort);
 - Ordenação por inserção direta (insertion sort);
 - Ordenação por permutação (bubble sort);
- Para avaliar a economia de tempo em métodos *in situ* é interessante avaliar o número de comparações e movimentações necessárias;

Ordenação por Seleção Direta

- Dentre os algoritmo de ordenação, o algoritmo de ordenação por seleção direta talvez seja o mais intuitivo;
- Seu princípio básico é:
 - Encontre a posição do menor elemento do vetor e troque com o elemento que está na primeira posição;
 - Em seguida encontre a posição do segundo menor e troque com a segunda posição;
 - ...
 - Encontre a posição do N-ésimo menor elemento e troque-o com o elemento da N-ésima posição;
 - ...
 - Faça isso até a penúltima posição do vetor

Exemplo de ordenação por seleção

valor	7	22	3	9	54	10	8	66	54	32
índice	0	1	2	3	4	5	6	7	8	9
	i									

índice menor	0
--------------	---

Exemplo de ordenação por seleção

valor	7	22	3	9	54	10	8	66	54	32
índice	0	1	2	3	4	5	6	7	8	9
	i	j								
índice menor		0								

Exemplo de ordenação por seleção

valor	7	22	3	9	54	10	8	66	54	32
índice	0	1	2	3	4	5	6	7	8	9
	i		j							

índice menor	2
--------------	---

Exemplo de ordenação por seleção

valor	7	22	3	9	54	10	8	66	54	32
índice	0	1	2	3	4	5	6	7	8	9
	i			j						

índice menor	2
--------------	---

Exemplo de ordenação por seleção

valor	7	22	3	9	54	10	8	66	54	32
índice	0	1	2	3	4	5	6	7	8	9
	i				j					

índice menor	2
--------------	---

Exemplo de ordenação por seleção

valor	7	22	3	9	54	10	8	66	54	32
índice	0	1	2	3	4	5	6	7	8	9
	i					j				

índice menor	2
--------------	---

Exemplo de ordenação por seleção

valor	7	22	3	9	54	10	8	66	54	32
índice	0	1	2	3	4	5	6	7	8	9
	i						j			

índice menor	2
--------------	---

Exemplo de ordenação por seleção

valor	7	22	3	9	54	10	8	66	54	32
índice	0	1	2	3	4	5	6	7	8	9
	i							j		

índice menor	2
--------------	---

Exemplo de ordenação por seleção

valor	7	22	3	9	54	10	8	66	54	32
índice	0	1	2	3	4	5	6	7	8	9
	i								j	

índice menor	2
--------------	---

Exemplo de ordenação por seleção

valor	7	22	3	9	54	10	8	66	54	32
índice	0	1	2	3	4	5	6	7	8	9
	i									j

índice menor	2
--------------	---

Exemplo de ordenação por seleção

Trocando elementos da posição 2 com posição da 0

valor	3	22	7	9	54	10	8	66	54	32
índice	0	1	2	3	4	5	6	7	8	9

índice menor	2
--------------	---

Exemplo de ordenação por seleção

valor	3	22	7	9	54	10	8	66	54	32
índice	0	1	2	3	4	5	6	7	8	9
		i								

índice menor	1
--------------	---

Exemplo de ordenação por seleção

valor	3	22	7	9	54	10	8	66	54	32
índice	0	1	2	3	4	5	6	7	8	9
		i	j							
índice menor		2								

Exemplo de ordenação por seleção

valor	3	22	7	9	54	10	8	66	54	32
índice	0	1	2	3	4	5	6	7	8	9
		i		j						
índice menor		2								

Exemplo de ordenação por seleção

valor	3	22	7	9	54	10	8	66	54	32
índice	0	1	2	3	4	5	6	7	8	9
		i			j					
índice menor		2								

Exemplo de ordenação por seleção

valor	3	22	7	9	54	10	8	66	54	32
índice	0	1	2	3	4	5	6	7	8	9
		i				j				
índice menor		2								

Exemplo de ordenação por seleção

valor	3	22	7	9	54	10	8	66	54	32
índice	0	1	2	3	4	5	6	7	8	9
		i					j			

índice menor	2
--------------	---

Exemplo de ordenação por seleção

valor	3	22	7	9	54	10	8	66	54	32
índice	0	1	2	3	4	5	6	7	8	9
		i						j		

índice menor	2
-----------------	---

Exemplo de ordenação por seleção

valor	3	22	7	9	54	10	8	66	54	32
índice	0	1	2	3	4	5	6	7	8	9
		i							j	

índice menor	2
--------------	---

Exemplo de ordenação por seleção

valor	3	22	7	9	54	10	8	66	54	32
índice	0	1	2	3	4	5	6	7	8	9
		i								j

índice menor	2
--------------	---

Exemplo de ordenação por seleção

Trocando elementos da posição 1 com da posição 2

valor	3	7	22	9	54	10	8	66	54	32
índice	0	1	2	3	4	5	6	7	8	9

índice menor	2
--------------	---

Exemplo de ordenação por seleção

valor	3	7	22	9	54	10	8	66	54	32
índice	0	1	2	3	4	5	6	7	8	9
			i							

índice menor	2
--------------	---

Exemplo de ordenação por seleção

valor	3	7	22	9	54	10	8	66	54	32
índice	0	1	2	3	4	5	6	7	8	9
			i	j						

índice menor	3
--------------	---

Exemplo de ordenação por seleção

valor	3	7	22	9	54	10	8	66	54	32
índice	0	1	2	3	4	5	6	7	8	9
			i		j					

índice menor	3
--------------	---

Exemplo de ordenação por seleção

valor	3	7	22	9	54	10	8	66	54	32
índice	0	1	2	3	4	5	6	7	8	9
			i			j				

índice menor	3
--------------	---

Exemplo de ordenação por seleção

valor	3	7	22	9	54	10	8	66	54	32
índice	0	1	2	3	4	5	6	7	8	9
			i				j			
índice menor	6									

Exemplo de ordenação por seleção

valor	3	7	22	9	54	10	8	66	54	32
índice	0	1	2	3	4	5	6	7	8	9
			i					j		

índice menor	6
--------------	---

Exemplo de ordenação por seleção

valor	3	7	22	9	54	10	8	66	54	32
índice	0	1	2	3	4	5	6	7	8	9
			i						j	

índice menor	6
-----------------	---

Exemplo de ordenação por seleção

valor	3	7	22	9	54	10	8	66	54	32
índice	0	1	2	3	4	5	6	7	8	9
			i							j

índice menor	6
--------------	---

Exemplo de ordenação por seleção

Trocando elementos da posição 2 com da posição 6

valor	3	7	8	9	54	10	22	66	54	32
índice	0	1	2	3	4	5	6	7	8	9

índice menor	6
--------------	---

Exemplo de ordenação por seleção

valor	3	7	8	9	54	10	22	66	54	32
índice	0	1	2	3	4	5	6	7	8	9
				i						

índice menor	3
-----------------	---

Exemplo de ordenação por seleção

valor	3	7	8	9	54	10	22	66	54	32
índice	0	1	2	3	4	5	6	7	8	9
				i	j					

índice menor	3
--------------	---

Exemplo de ordenação por seleção

valor	3	7	8	9	54	10	22	66	54	32
índice	0	1	2	3	4	5	6	7	8	9
				i		j				

índice menor	3
--------------	---

Exemplo de ordenação por seleção

valor	3	7	8	9	54	10	22	66	54	32
índice	0	1	2	3	4	5	6	7	8	9
				i			j			

índice menor	3
--------------	---

Exemplo de ordenação por seleção

valor	3	7	8	9	54	10	22	66	54	32
índice	0	1	2	3	4	5	6	7	8	9
				i				j		

índice menor	3
--------------	---

Exemplo de ordenação por seleção

valor	3	7	8	9	54	10	22	66	54	32
índice	0	1	2	3	4	5	6	7	8	9
				i					j	

índice menor	3
-----------------	---

Exemplo de ordenação por seleção

valor	3	7	8	9	54	10	22	66	54	32
índice	0	1	2	3	4	5	6	7	8	9
				i						j

índice menor	3
--------------	---

Exemplo de ordenação por seleção

Trocando elemento da posição 3 com da posição 3

valor	3	7	8	9	54	10	22	66	54	32
índice	0	1	2	3	4	5	6	7	8	9
				i						j

índice menor	3
--------------	---

Exemplo de ordenação por seleção

valor	3	7	8	9	54	10	22	66	54	32
índice	0	1	2	3	4	5	6	7	8	9
					i					

índice menor	4
--------------	---

Exemplo de ordenação por seleção

valor	3	7	8	9	54	10	22	66	54	32
índice	0	1	2	3	4	5	6	7	8	9
					i	j				

índice menor	5
--------------	---

Exemplo de ordenação por seleção

valor	3	7	8	9	54	10	22	66	54	32
índice	0	1	2	3	4	5	6	7	8	9
					i		j			

índice menor	5
--------------	---

Exemplo de ordenação por seleção

valor	3	7	8	9	54	10	22	66	54	32
índice	0	1	2	3	4	5	6	7	8	9
					i			j		

índice menor	5
-----------------	---

Exemplo de ordenação por seleção

valor	3	7	8	9	54	10	22	66	54	32
índice	0	1	2	3	4	5	6	7	8	9
					i				j	

índice menor	5
--------------	---

Exemplo de ordenação por seleção

valor	3	7	8	9	54	10	22	66	54	32
índice	0	1	2	3	4	5	6	7	8	9
					i					j

índice menor	5
--------------	---

Exemplo de ordenação por seleção

Trocando elemento da posição 4 com da posição 5

valor	3	7	8	9	10	54	22	66	54	32
índice	0	1	2	3	4	5	6	7	8	9

índice menor	5
--------------	---

Exemplo de ordenação por seleção

valor	3	7	8	9	10	54	22	66	54	32
índice	0	1	2	3	4	5	6	7	8	9
						i				

índice menor	5
-----------------	---

Exemplo de ordenação por seleção

valor	3	7	8	9	10	54	22	66	54	32
índice	0	1	2	3	4	5	6	7	8	9
						i	j			

índice menor	6
--------------	---

Exemplo de ordenação por seleção

valor	3	7	8	9	10	54	22	66	54	32
índice	0	1	2	3	4	5	6	7	8	9
						i		j		

índice menor	6
-----------------	---

Exemplo de ordenação por seleção

valor	3	7	8	9	10	54	22	66	54	32
índice	0	1	2	3	4	5	6	7	8	9
						i			j	

índice menor	6
-----------------	---

Exemplo de ordenação por seleção

valor	3	7	8	9	10	54	22	66	54	32
índice	0	1	2	3	4	5	6	7	8	9
						i				j

índice menor	6
--------------	---

Exemplo de ordenação por seleção

Trocando elemento da posição 5 com da posição 6

valor	3	7	8	9	10	22	54	66	54	32
índice	0	1	2	3	4	5	6	7	8	9

índice menor	6
--------------	---

Exemplo de ordenação por seleção

[illegible]

índice menor	6
--------------	---

Exemplo de ordenação por seleção

valor	3	7	8	9	10	22	54	66	54	32
índice	0	1	2	3	4	5	6	7	8	9
							i	j		

índice menor	6
--------------	---

Exemplo de ordenação por seleção

valor	3	7	8	9	10	22	54	66	54	32
índice	0	1	2	3	4	5	6	7	8	9
							i		j	

índice menor	6
-----------------	---

Exemplo de ordenação por seleção

[illegible]

índice menor	9
--------------	---

Exemplo de ordenação por seleção

Trocando elemento da posição 6 com da posição 9

valor	3	7	8	9	10	22	32	66	54	54
índice	0	1	2	3	4	5	6	7	8	9

índice menor	6
--------------	---

Exemplo de ordenação por seleção

valor	3	7	8	9	10	22	32	66	54	54
índice	0	1	2	3	4	5	6	7	8	9
								i		

índice menor	7
-----------------	---

Exemplo de ordenação por seleção

valor	3	7	8	9	10	22	32	66	54	54
índice	0	1	2	3	4	5	6	7	8	9
								i	j	

índice menor	8
-----------------	---

Exemplo de ordenação por seleção

valor	3	7	8	9	10	22	32	66	54	54
índice	0	1	2	3	4	5	6	7	8	9
								i		j

índice menor	8
--------------	---

Exemplo de ordenação por seleção

Trocando elemento da posição 7 com o da posição 8

valor	3	7	8	9	10	22	32	54	66	54
índice	0	1	2	3	4	5	6	7	8	9

índice menor	8
--------------	---

Exemplo de ordenação por seleção

valor	3	7	8	9	10	22	32	54	66	54
índice	0	1	2	3	4	5	6	7	8	9
									i	

índice menor	8
-----------------	---

Exemplo de ordenação por seleção

valor	3	7	8	9	10	22	32	54	66	54
índice	0	1	2	3	4	5	6	7	8	9
									i	j

índice menor	9
--------------	---

Exemplo de ordenação por seleção

Trocando elemento da posição 8 com o da posição 9

valor	3	7	8	9	10	22	32	54	54	66
índice	0	1	2	3	4	5	6	7	8	9

índice menor	8
--------------	---

Ordenação por seleção

- É um método interessante para a ordenação de poucos valores;
- O algoritmo tem um aspecto negativo que é não tomar proveito de vetores parcialmente ordenados;
 - Alguns algoritmos possuem essa vantagem e em seus melhores casos reduzem significativamente a quantidade de comparações e trocas!
- O algoritmo de ordenação por seleção possui no pior caso uma complexidade de $O(n^2)$;

Implementação

```
void selection(int v[], int n){  
    /**/  
}
```


Implementação

```
void selection(int v[], int n){
    int i,j, idx_menor;
    for(i=0;i<n-1;i++){
        idx_menor=i;
        for(j=i+1;j<n;j++){
            if(v[j]<v[idx_menor]){
                idx_menor=j;
            }
        }
        trocar(&v[i],&v[idx_menor]);
    }
}
```

Ordenação por inserção direta

- Outra maneira simples de se ordenar valores é o método da inserção direta;
- Método comumente usado por jogadores de cartas;
 - A cada carta nova recebida, tal carta é inserida na sua devida posição levando o critério de ordem;
 - Desse modo, as cartas sempre estarão ordenadas na mão do jogador;

Ordenação por inserção direta

- Divide-se um vetor de n elementos em duas sequências
 - Uma sequência destino ($a_1 \dots a_{i-1}$);
 - As cartas que estão na mão do jogador;
 - Uma sequência fonte ($a_i \dots a_n$);
 - O baralho com as novas cartas que serão recebidas;
- Todos os elementos que estão na sequência destino estão ordenados;
- Os elementos da sequência fonte tem a ordem desconhecida;

Ordenação por inserção direta

- A sequência destino é iniciada com um elemento;
 - Como ele é único, a sequência é iniciada ordenada
- A cada iteração de um laço, o primeiro elemento da sequência fonte (o *i*-ésimo elemento do vetor) é “retirado” da sequência fonte e inserido em seu devido lugar na sequência destino;
 - Desse modo, a sequência destino permanece ordenada
- O algoritmo termina quando todos os elementos forem inseridos na sequência destino;

Ordenação por inserção direta

- Como colocar o elemento em sua devida posição?

Ordenação por inserção direta

- Como colocar o elemento x em sua devida posição?
 - Salve x em uma variável temporária
 - Coloque todos elementos maiores que x uma posição à frente no vetor;
 - Coloque x na posição do menor elemento que foi “arrastado”;

Execução do Algoritmo de inserção

[illegible]

Execução do Algoritmo de inserção

x	2
---	---

valor	5	2	1	4	3	6	9	7	8
índice	0	1	2	3	4	6	7	8	9
		i							

Execução do Algoritmo de inserção

x	2
---	---

valor	5	2	1	4	3	6	9	7	8
índice	0	1	2	3	4	6	7	8	9
		i j							

Execução do Algoritmo de inserção

x	2
---	---

valor	5	5	1	4	3	6	9	7	8
índice	0	1	2	3	4	6	7	8	9
	j	i							

Execução do Algoritmo de inserção

x	2
---	---

valor	2	5	1	4	3	6	9	7	8
índice	0	1	2	3	4	6	7	8	9
	j	i							

Execução do Algoritmo de inserção

x	1
---	---

valor	2	5	1	4	3	6	9	7	8
índice	0	1	2	3	4	6	7	8	9
			ij						

Execução do Algoritmo de inserção

x	1
---	---

valor	2	5	5	4	3	6	9	7	8
índice	0	1	2	3	4	6	7	8	9
		j	i						

Execução do Algoritmo de inserção

x	1
---	---

valor	2	2	5	4	3	6	9	7	8
índice	0	1	2	3	4	6	7	8	9
	j		i						

Execução do Algoritmo de inserção

x	1
---	---

valor	1	2	5	4	3	6	9	7	8
índice	0	1	2	3	4	6	7	8	9
	j		i						

Execução do Algoritmo de inserção

x	4
---	---

valor	1	2	5	4	3	6	9	7	8
índice	0	1	2	3	4	6	7	8	9
				i j					

Execução do Algoritmo de inserção

x	4
---	---

valor	1	2	5	5	3	6	9	7	8
índice	0	1	2	3	4	6	7	8	9
			j	i					

Execução do Algoritmo de inserção

x	4
---	---

valor	1	2	4	5	3	6	9	7	8
índice	0	1	2	3	4	6	7	8	9
			j	i					

Note que o valor de dois não é copiado, pois $\text{vetor}[j-1] \leq x$

Execução do Algoritmo de inserção

x	3
---	---

valor	1	2	4	5	3	6	9	7	8
índice	0	1	2	3	4	6	7	8	9
					ij				

Execução do Algoritmo de inserção

x	3
---	---

valor	1	2	4	5	5	6	9	7	8
índice	0	1	2	3	4	6	7	8	9
				j	i				

Execução do Algoritmo de inserção

x	3
---	---

valor	1	2	4	4	5	6	9	7	8
índice	0	1	2	3	4	6	7	8	9
			j		i				

Execução do Algoritmo de inserção

x	3
---	---

valor	1	2	3	4	5	6	9	7	8
índice	0	1	2	3	4	6	7	8	9
			j		i				

Execução do Algoritmo de inserção

x	6
---	---

valor	1	2	3	4	5	6	9	7	8
índice	0	1	2	3	4	6	7	8	9
						ij			

Execução do Algoritmo de inserção

x	6
---	---

valor	1	2	3	4	5	6	9	7	8
índice	0	1	2	3	4	6	7	8	9
						i j			

Como logo na primeira iteração da estrutura de repetição que decrementa j $\text{vetor}[j-1] \leq x$.

Execução do Algoritmo de inserção

x	9
---	---

[illegible]

Execução do Algoritmo de inserção

x	7
---	---

[illegible]

Execução do Algoritmo de inserção

x	7
---	---

[illegible]

Execução do Algoritmo de inserção

x	7
---	---

[illegible]

Execução do Algoritmo de inserção

x	8
---	---

[illegible]

Execução do Algoritmo de inserção

x	8
---	---

[illegible]

Execução do Algoritmo de inserção

x	8
---	---

[illegible]

Ordenação por inserção direta

- A ordenação por inserção direta funciona bem para vetores que estão quase ordenados;
- É também um bom método para adicionar poucos elementos a um vetor já ordenado;
- O algoritmo de ordenação por inserção direta possui no pior caso uma complexidade de $O(n^2)$;

Implementação

```
void insertion(int v[],int n){  
    /*algoritmo*/  
}
```

Implementação

```
void insertion(int v[],int n){  
    int i,j,x;  
    for(i=1;i<n;i++){  
        x=v[i];  
        j=i;  
        while(v[j-1]>x && j>0){  
            v[j]=v[j-1];  
            j--;  
        }  
        v[j]=x;  
    }  
}
```

Ordenação por Permutação

- Alguns algoritmos tem como sua principal característica a permutação entre elementos;
- Tais métodos são também chamados de ordenação por permutação;
- Nesses métodos a permutação ocorre entre elementos vizinhos;

Ordenação por Permutação

- O método mais conhecido que faz permutação entre dois vizinhos é o Bubble Sort;
- O Bubble Sort age da seguinte maneira:
 - Percorra o vetor diversas vezes de maneira sequencial;
 - Permute vizinhos que estão fora de ordem;

Ordenação por Permutação

- Pode-se fazer uma analogia a um tanque cheio de água e bolhas nesse tanque;
 - Sendo cada um dos elementos uma bolha, e sua densidade correspondente ao valor do elemento, cada vez que for feita uma “varredura” no vetor, obteremos a ascensão de uma bolha para seu nível apropriado;

Ordenação por Permutação

- Após a primeira “varredura” no vetor, o maior elemento estará na última posição;
- Após a segunda “varredura” no vetor, o segundo maior elemento estará na penúltima posição;
- E assim por diante;

Execução do Bubble Sort

i	0
---	---

valor	50	25	12	7	9
índice	0	1	2	3	4

Execução do Bubble Sort

Trocar 50 com 25

i	0
---	---

valor	50	25	12	7	9
índice	0	1	2	3	4
	j	j+1			

Execução do Bubble Sort

i	0
---	---

valor	25	50	12	7	9
índice	0	1	2	3	4
	j	j+1			

Execução do Bubble Sort

Trocar 50 com 12

i	0
---	---

valor	25	50	12	7	9
índice	0	1	2	3	4
		j	j+1		

Execução do Bubble Sort

i	0
---	---

valor	25	12	50	7	9
índice	0	1	2	3	4
		j	j+1		

Execução do Bubble Sort

Trocar 50 com 7

i	0
---	---

valor	25	12	50	7	9
índice	0	1	2	3	4
			j	j+1	

Execução do Bubble Sort

i	0
---	---

valor	25	12	7	50	9
índice	0	1	2	3	4
			j	j+1	

Execução do Bubble Sort

Trocar 50 com 9

i	0
---	---

valor	25	12	7	50	9
índice	0	1	2	3	4
				j	j+1

Execução do Bubble Sort

i	0
---	---

valor	25	12	7	9	50
índice	0	1	2	3	4
				j	j+1

Execução do Bubble Sort

Trocar 25 com 12

i	1
---	---

valor	25	12	7	9	50
índice	0	1	2	3	4
	j	j+1			

Execução do Bubble Sort

i	1
---	---

valor	12	25	7	9	50
índice	0	1	2	3	4
	j	j+1			

Execução do Bubble Sort

Trocar 25 com 7

i	1
---	---

valor	12	25	7	9	50
índice	0	1	2	3	4
		j	j+1		

Execução do Bubble Sort

i	1
---	---

valor	12	7	25	9	50
índice	0	1	2	3	4
		j	j+1		

Execução do Bubble Sort

Trocar 25 com 9

i	1
---	---

valor	12	7	25	9	50
índice	0	1	2	3	4
			j	j+1	

Execução do Bubble Sort

i	1
---	---

valor	12	7	9	25	50
índice	0	1	2	3	4
			j	j+1	

Execução do Bubble Sort

Trocar 12 com 7

i	2
---	---

valor	12	7	9	25	50
índice	0	1	2	3	4
	j	j+1			

Execução do Bubble Sort

i	2
---	---

valor	7	12	9	25	50
índice	0	1	2	3	4
	j	j+1			

Execução do Bubble Sort

Trocar 12 com 9

i	2
---	---

valor	7	12	9	25	50
índice	0	1	2	3	4
		j	j+1		

Execução do Bubble Sort

i	2
---	---

valor	7	9	12	25	50
índice	0	1	2	3	4
		j	j+1		

Execução do Bubble Sort

i	3
---	---

valor	7	9	12	25	50
índice	0	1	2	3	4
	j	j+1			

Execução do Bubble Sort

valor	7	9	12	25	50
índice	0	1	2	3	4

Implementação

```
void bubble(int v[],int n){  
    /*algoritmo*/  
}
```

Implementação

```
void bubble(int v[],int n){
    int i,j;
    for(i=0;i<n-1;i++){
        for(j=0;j<(n-1)-i;j++){
            if(v[j]> v[j+1]){
                trocar(&v[j],&v[j+1]);
            }
        }
    }
}
```

Melhorando o Bubble Sort

- No Bubble Sort, quando uma varredura é realizada no vetor e não houveram permutações o vetor está ordenado;
 - Pode-se usar tal informação para melhorar o algoritmo;
 - Basta utilizar uma variável para controlar tal situação;
- Um outro detalhe do bubble sort é que, o vetor seja ordenado em ordem não crescente, basta inverter o sinal da comparação;
- O algoritmo bubble sort, tanto em sua versão comum, quanto em sua versão melhorada, possui no pior caso uma complexidade de $O(n^2)$;

Ordenação por Permutação

- Uma variante do Bubblesort é o Bubblesort bidirecional;
 - Também chamado de Cocktailsort;
- Usa a mesma estratégia do Bubblesort de trocar vizinhos em várias varreduras, sua diferença é que ao invés de levar somente o maior elemento para a última posição no laço de repetição externo, ele também trás o menor elemento para a primeira posição;

Ordenação

- Animações para os algoritmos:
 - Bubble sort;
 - Insertion sort;
 - Selection sort;

<https://visualgo.net/en/sorting>

Exercícios

1. Implemente a versão melhorada do algoritmo bubble sort.
2. Implemente o algoritmo cocktail sort.

Referências

Robert Sedgewick. 1990. *Algorithms in C*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.

Wirth, Niklaus. Algoritmos e Estruturas de Dados, Editora LTC