

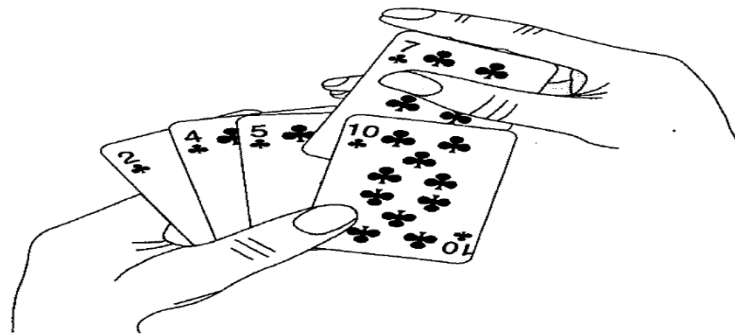
INSERTION SORT

Estrutura de dados

INSERTION SORT

Algoritmo eficiente para ordenar uma lista com poucos elementos;

O método funciona do mesmo modo que muitas pessoas ordenam suas cartas em um jogo de cartas.



Estrutura de dados

INSERTION SORT

- A cada iteração é considerado um índice i tal que os elementos a esquerda de i já se encontram ordenados.
- O item de índice i será inserido na posição correta considerando apenas os elementos à sua esquerda.
- Percorre-se, então, os itens já ordenados da direita para a esquerda comparando cada item com o item a ser inserido: enquanto o item a ser inserido for menor, o item que está sendo comparado será deslocado à direita.

Estrutura de dados

INSERTION SORT

Chaves Iniciais:

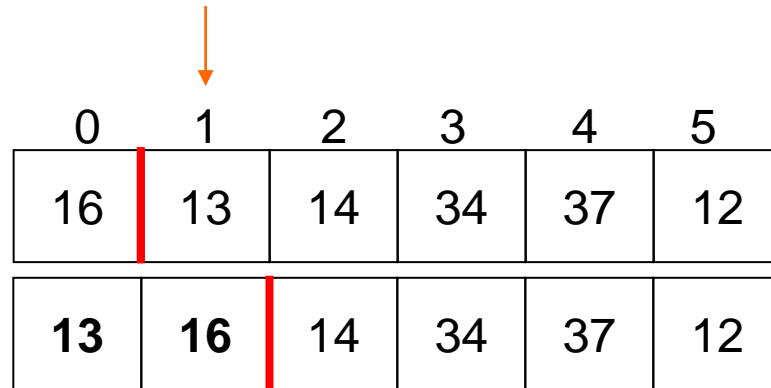
i=1:

0	1	2	3	4	5
16	13	14	34	37	12

Estrutura de dados

INSERTION SORT

Chaves Iniciais:



0	1	2	3	4	5
16	13	14	34	37	12
13	16	14	34	37	12

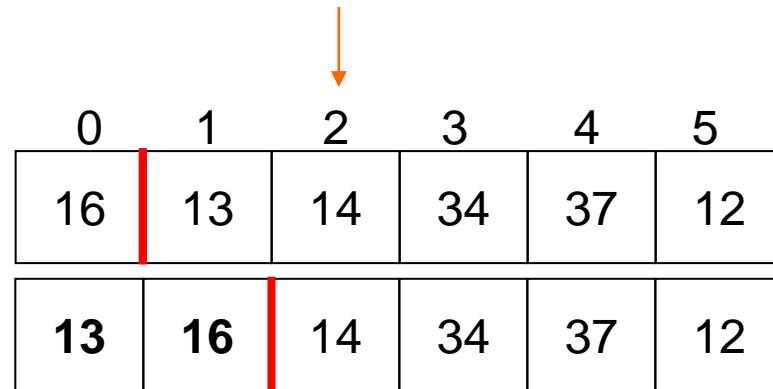
i=1:

i=2:

Estrutura de dados

INSERTION SORT

Chaves Iniciais:



0	1	2	3	4	5
16	13	14	34	37	12
13	16	14	34	37	12

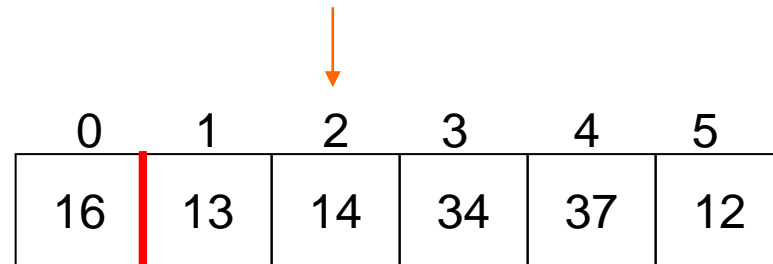
i=1:

i=2:

Estrutura de dados

INSERTION SORT

Chaves Iniciais:



0	1	2	3	4	5
16	13	14	34	37	12

i=1:

13	16	14	34	37	12
-----------	-----------	----	----	----	----

i=2:


13	14	16	34	37	12
-----------	-----------	-----------	----	----	----

i=3:

Estrutura de dados

INSERTION SORT

Chaves Iniciais:



0	1	2	3	4	5
16	13	14	34	37	12

i=1:

13	16	14	34	37	12
-----------	-----------	----	----	----	----

i=2:


13	14	16	34	37	12
-----------	-----------	-----------	----	----	----

i=3:

Estrutura de dados

INSERTION SORT

Chaves Iniciais:



0	1	2	3	4	5
16	13	14	34	37	12

i=1:

13	16	14	34	37	12
-----------	-----------	----	----	----	----

i=2:

13	14	16	34	37	12
-----------	-----------	-----------	----	----	----

i=3:


13	14	16	34	37	12
-----------	-----------	-----------	-----------	----	----

i=4:

Estrutura de dados

INSERTION SORT

Chaves Iniciais:



0	1	2	3	4	5
16	13	14	34	37	12

i=1:

13	16	14	34	37	12
-----------	-----------	----	----	----	----

i=2:

13	14	16	34	37	12
-----------	-----------	-----------	----	----	----

i=3:


13	14	16	34	37	12
-----------	-----------	-----------	-----------	----	----

i=4:

Estrutura de dados

INSERTION SORT

Chaves Iniciais:



0	1	2	3	4	5
16	13	14	34	37	12

i=1:

13	16	14	34	37	12
-----------	-----------	----	----	----	----

i=2:

13	14	16	34	37	12
-----------	-----------	-----------	----	----	----

i=3:

13	14	16	34	37	12
-----------	-----------	-----------	-----------	----	----

i=4:

13	14	16	34	37	12
-----------	-----------	-----------	-----------	-----------	----

i=5:

Estrutura de dados

INSERTION SORT

Chaves Iniciais:

0	1	2	3	4	5
16	13	14	34	37	12

i=1:

13	16	14	34	37	12
----	----	----	----	----	----

i=2:

13	14	16	34	37	12
----	----	----	----	----	----

i=3:

13	14	16	34	37	12
----	----	----	----	----	----

i=4:

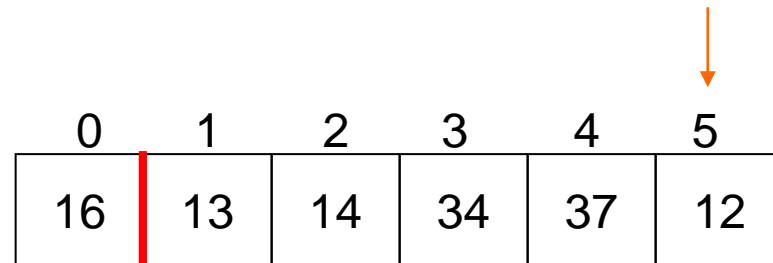
13	14	16	34	37	12
----	----	----	----	----	----

i=5:

Estrutura de dados

INSERTION SORT

Chaves Iniciais:



0	1	2	3	4	5
16	13	14	34	37	12

i=1:

13	16	14	34	37	12
----	----	----	----	----	----

i=2:

13	14	16	34	37	12
----	----	----	----	----	----

i=3:

13	14	16	34	37	12
----	----	----	----	----	----

i=4:

13	14	16	34	37	12
----	----	----	----	----	----

i=5:

12	13	14	16	34	37
----	----	----	----	----	----

Estrutura de dados

INSERTION SORT

aux = 12 i=5

Exemplo de como cada elemento é passado para o lado ordenado do vetor.

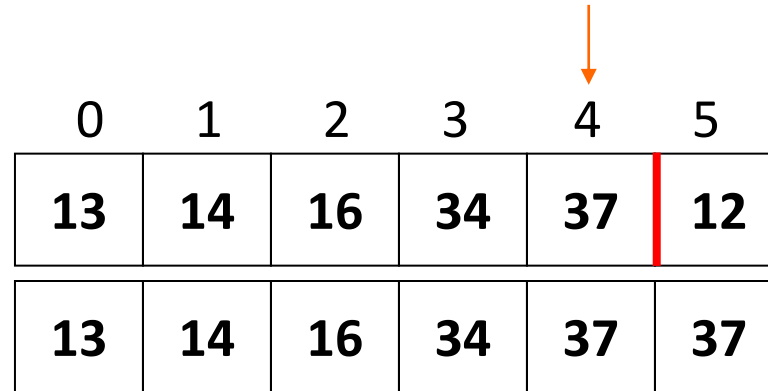
0	1	2	3	4	5
13	14	16	34	37	12

Estrutura de dados

INSERTION SORT

aux = 12 i=5

j=4

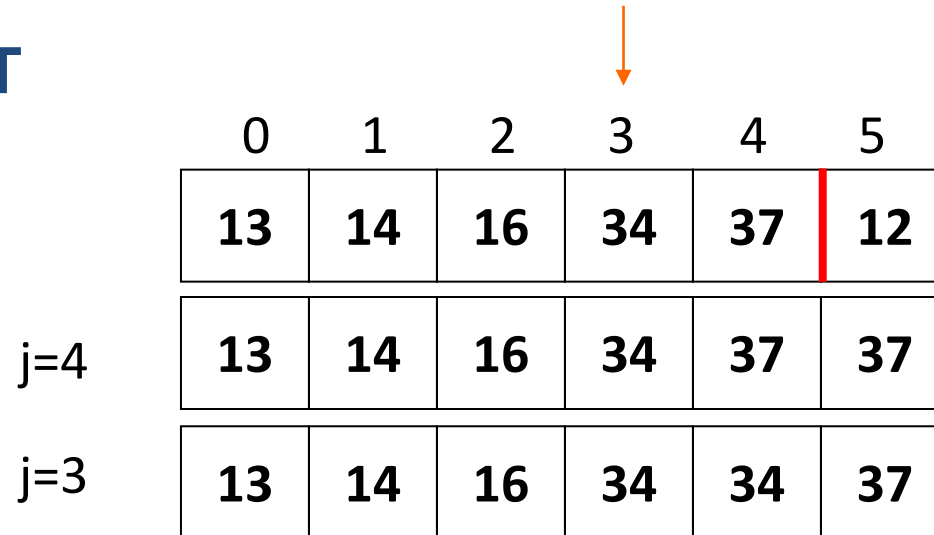


0	1	2	3	4	5
13	14	16	34	37	12
13	14	16	34	37	37

Estrutura de dados

INSERTION SORT

aux = 12 i=5



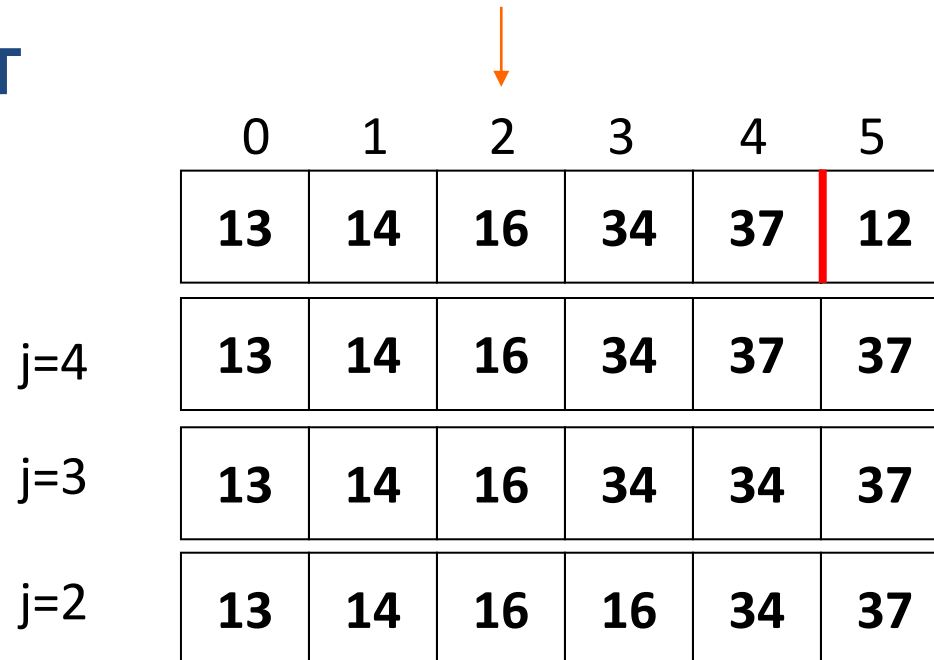
The diagram illustrates the insertion sort process. It shows three rows of an array with indices 0 to 5. The first row shows the initial state with values [13, 14, 16, 34, 37, 12]. An orange arrow points to the element 12 at index 5. A red vertical line is placed between index 4 and 5. The second row, labeled j=4, shows the state after shifting elements from index 4 to index 5, resulting in [13, 14, 16, 34, 37, 37]. The third row, labeled j=3, shows the state after shifting elements from index 3 to index 4, resulting in [13, 14, 16, 34, 34, 37].

0	1	2	3	4	5
13	14	16	34	37	12
j=4	13	14	16	34	37
j=3	13	14	16	34	37

Estrutura de dados

INSERTION SORT

aux = 12 i=5

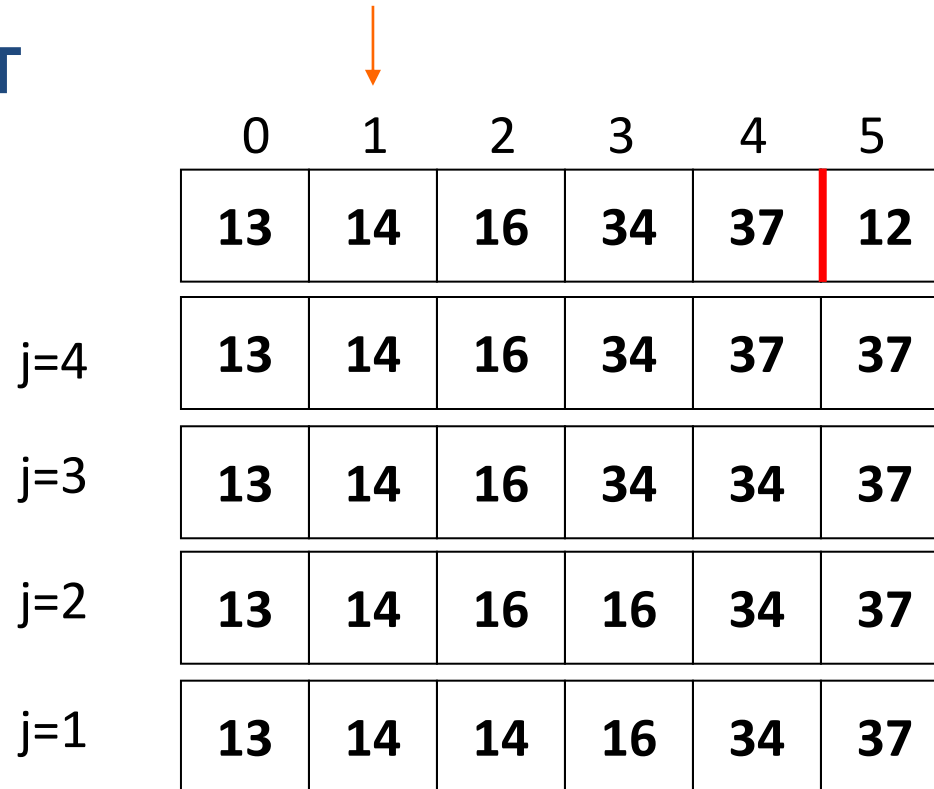


	0	1	2	3	4	5
	13	14	16	34	37	12
j=4	13	14	16	34	37	37
j=3	13	14	16	34	34	37
j=2	13	14	16	16	34	37

Estrutura de dados

INSERTION SORT

aux = 12 i=5

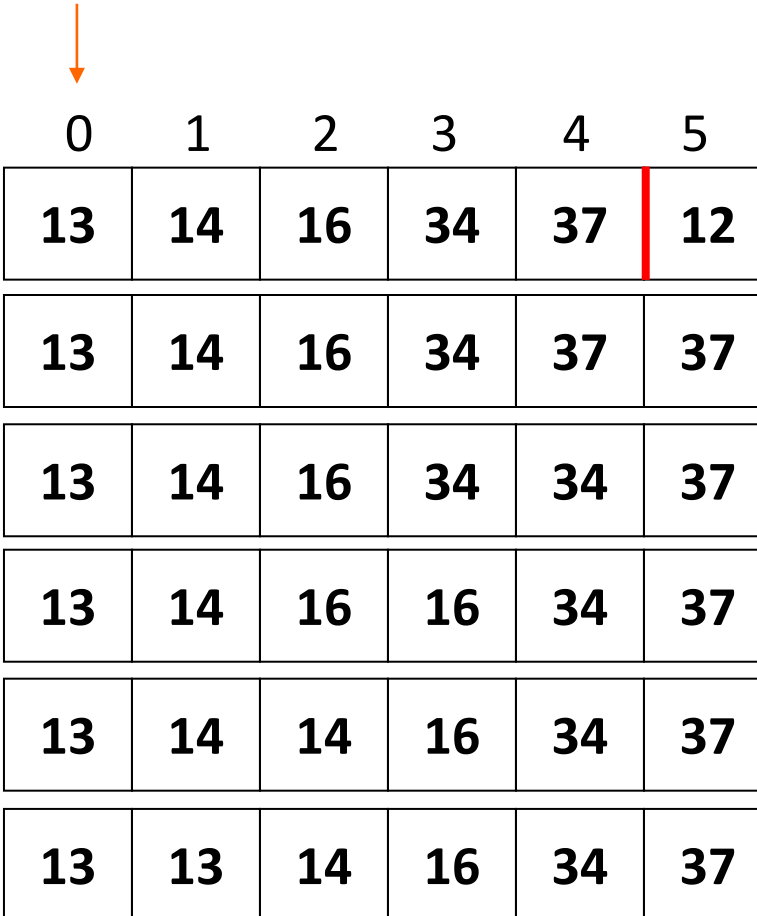


	0	1	2	3	4	5
	13	14	16	34	37	12
j=4	13	14	16	34	37	37
j=3	13	14	16	34	34	37
j=2	13	14	16	16	34	37
j=1	13	14	14	16	34	37

Estrutura de dados

INSERTION SORT

aux = 12 i=5



	0	1	2	3	4	5
	13	14	16	34	37	12
j=4	13	14	16	34	37	37
j=3	13	14	16	34	34	37
j=2	13	14	16	16	34	37
j=1	13	14	14	16	34	37
j=0	13	13	14	16	34	37

Estrutura de dados

INSERTION SORT



aux = 12 i=5

	0	1	2	3	4	5
	13	14	16	34	37	12
j=4	13	14	16	34	37	37
j=3	13	14	16	34	34	37
j=2	13	14	16	16	34	37
j=1	13	14	14	16	34	37
j=0	13	13	14	16	34	37
j=-1	12	13	14	16	34	37

Estrutura de dados

INSERTION SORT

```
void insercaoDireta(int *vet, int n){
    int i, j, aux;
    for (i=1; i<n; i++) {
        aux = vet[i];
        j = i -1;
        while (aux< vet[j] && j>=0){
            vet[j+1] = vet[j];
            j = j-1;
        }
        vet[j+1] = aux;
    }
}
```

Estrutura de dados

INSERTION SORT - EXERCÍCIO

Para o método de ordenação de seleção, apresente a ordenação do seguinte vetor. Mostre a sequência de todas as etapas executadas (comparações e trocas).

$$V = \{11; 9; 7; 5; 3; 1\}$$