

Bubble Sort e Insertion Sort

Prof. Dr. Eleandro Maschio
Tecnologia em Sistemas para Internet
Câmpus Guarapuava
Universidade Tecnológica Federal do Paraná (UTFPR)

Tipos de Métodos de Ordenação

Existem três métodos gerais para ordenar matrizes unidimensionais:

- Por **seleção**.
- Por **troca**.
- Por **inserção**.

Ordenação por Trocas

(1) Espalhe as cartas na mesa com os números voltados para cima.

(2) **Troque** as cartas fora de ordem até que **todo o baralho esteja ordenado**.

As trocas devem ser feitas de uma carta por outra, nunca inserindo cartas entre as outras ou selecionando cartas para a troca.

Portanto, geralmente as trocas envolvem a posição imediatamente anterior ou posterior à carta corrente.

Bubble Sort

- O método de ordenação mais conhecido (e mais difamado) é a **ordenação bolha**.
- A popularidade deve-se ao nome fácil e à implementação simples.
- Porém, é um dos **piores métodos** de ordenação já concebidos.
- Trata-se de uma ordenação por troca. Envolve **repetidas comparações** e, se necessária, a troca de dois **elementos adjacentes**.
- Os elementos são como bolhas em um tanque de água: cada uma procura o seu próprio nível.

Bubble Sort

Lógica de Implementação

- Compare o **primeiro elemento** com o **segundo**. Caso o primeiro seja **maior, troque** com o segundo.
- Depois, compare o **segundo** elemento com o **terceiro**. Caso o segundo seja **maior, troque** com o terceiro.
- Continue desta forma até comparar o **penúltimo** elemento com o **último**. Caso o penúltimo seja **maior, troque** com o último.

Com isso, na **última posição** estará armazenado o **maior elemento**.

Bubble Sort

Lógica de Implementação (...)

- Novamente, compare o **primeiro** elemento com o **segundo**. Caso o primeiro seja **maior**, **troque** com o segundo.
- Compare o **segundo** elemento com o **terceiro**. Caso o segundo seja maior, **troque** com o **terceiro**.
- Continue até comparar o **antepenúltimo** elemento com o **penúltimo**. Caso o antepenúltimo seja maior, **troque** com o **penúltimo**.

Com isso, na **penúltima** posição estará armazenado o **segundo maior elemento**. Note que **não se mexe na última posição**.

Bubble Sort

Lógica de Implementação (...)

- Continue com o método até que na **terceira posição** esteja o **terceiro maior valor**.
- Então compare a **primeira** e a **segunda** posição para saber qual será o **menor** e o **segundo menor** valor.
- Feito isto, a sequência está ordenada.

Bubble Sort, Comparações e Trocas

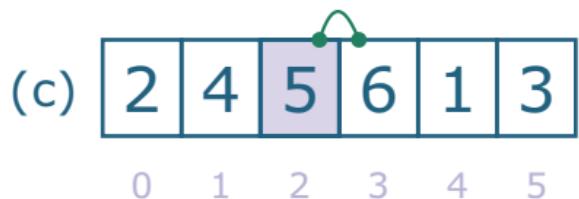
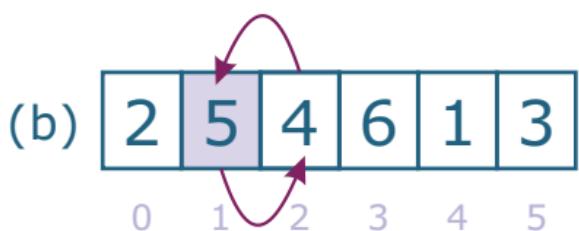
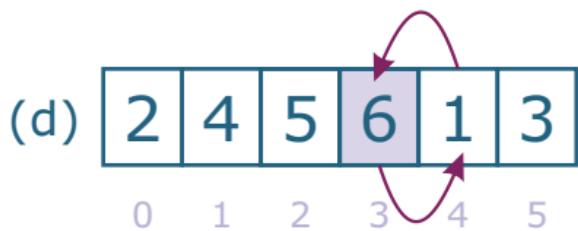
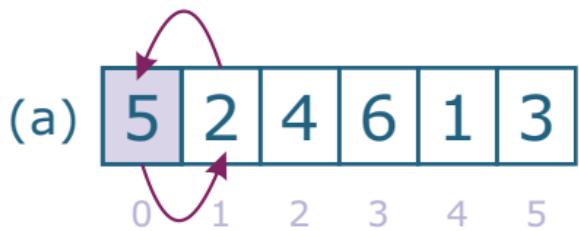
- Note que **inúmeras trocas e comparações desnecessárias** são feitas.
- Por este motivo, autores como Herbert Schildt (*C Completo e Total*) o famigeraram como **O Demônio das Trocas**.

Dinâmica de Funcionamento

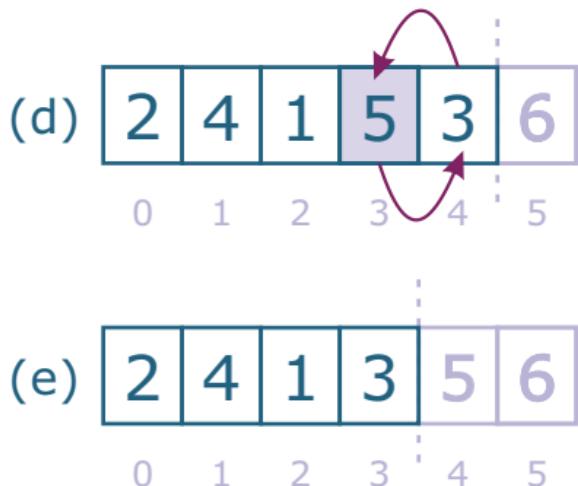
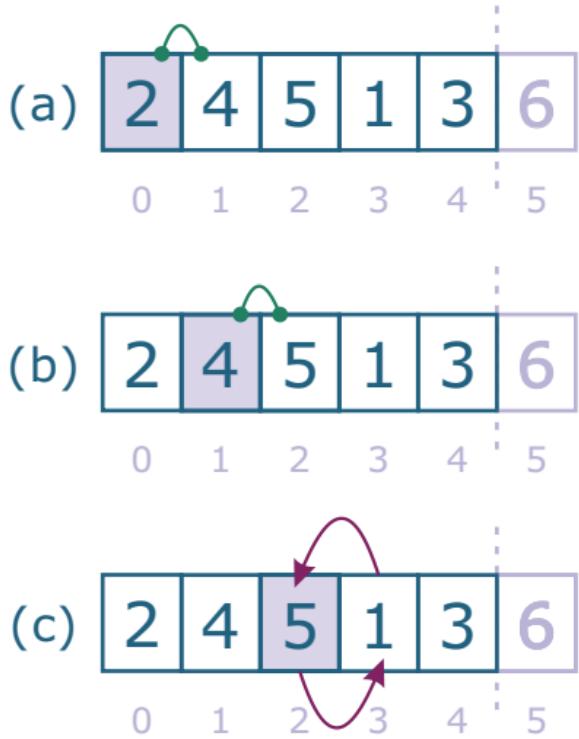
Considere a matriz unidimensional.

| | | | | | |
|---|---|---|---|---|---|
| 5 | 2 | 4 | 6 | 1 | 3 |
| 0 | 1 | 2 | 3 | 4 | 5 |

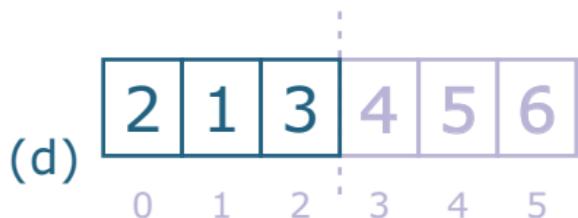
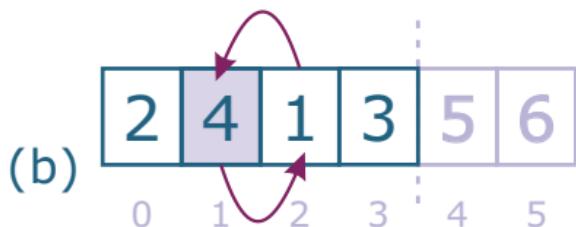
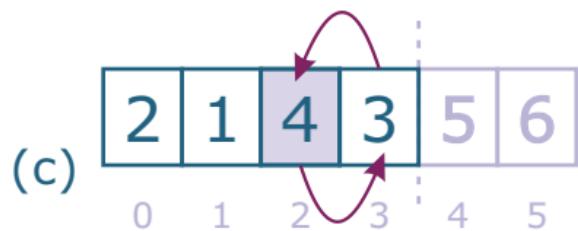
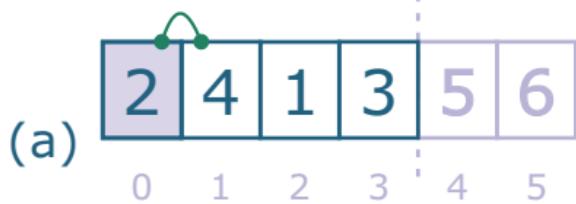
Dinâmica de Funcionamento



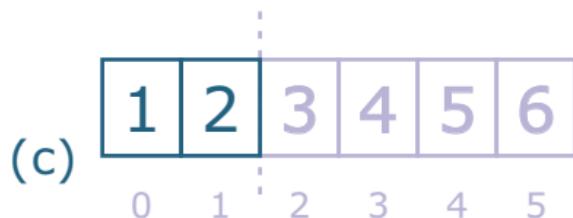
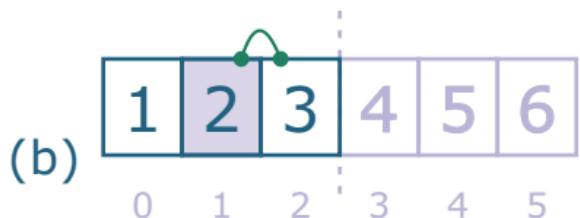
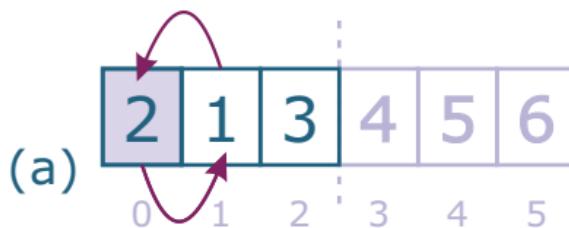
Dinâmica de Funcionamento



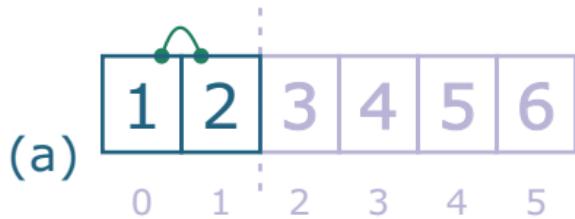
Dinâmica de Funcionamento



Dinâmica de Funcionamento



Dinâmica de Funcionamento



Implementação

Laço externo

Laço interno

Comparação e troca dos elementos alocados em posições $\leq TAM-i$.

Fazer com que o i -ézimo maior elemento seja alocado na posição $TAM-i$.

Implementação

- É possível implementar uma dinâmica inversa para o Bubble Sort.
- Ao invés de levar os maiores elementos para o final da matriz, pode-se procurar trazer os **menores** elementos para o **início**.
- Para tal, o percorrimento deve ser feito de **trás para a frente**.
- O método continua sendo o Bubble Sort.

Ordenação por Inserção

- (1) Inicie com a **mão esquerda vazia** e as cartas viradas com a face para baixo da mesa.
- (2) **Remova** uma carta de cada vez da mesa, inserindo-a na **posição correta** na mão esquerda.

Para encontrar a posição correta de uma carta, **compare** com cada uma das cartas que já estão na mão, da direita para a esquerda.

Em cada instante, as cartas seguradas na mão esquerda estão **ordenadas** (cartas que eram originalmente as superiores da pilha da mesa).

As cartas estarão ordenadas quando não houver mais cartas na **pilha**.

Ordenação por Inserção

- Repare que a ordenação por inserção funciona da maneira como muitas pessoas ordenam as cartas em um jogo de baralho.
- A **ordenação por inserção** mostra-se eficiente para ordenar um pequeno número de elementos.

Insertion Sort

Lógica de Implementação com Matriz Auxiliar

- Crie uma matriz auxiliar de mesmo tamanho.
- Insira o **primeiro** elemento da sequência na **primeira posição** da matriz auxiliar.
- Então, insira o segundo elemento na matriz auxiliar.

Insira-o na ordem **correta** com relação ao primeiro elemento. Para isto, pode ser necessária a realocação do primeiro elemento. Não deixe lacunas entre elementos.

- O processo continua até que todos os elementos tenham sido inseridos corretamente na matriz auxiliar. Sem é feita realocação quando necessário.

Dinâmica de Funcionamento

Considere a matriz unidimensional.

| | | | | | |
|---|---|---|---|---|---|
| 5 | 2 | 4 | 6 | 1 | 3 |
| 0 | 1 | 2 | 3 | 4 | 5 |

Dinâmica de Funcionamento

| | | | | | |
|---|---|---|---|---|---|
| 5 | 2 | 4 | 6 | 1 | 3 |
| 0 | 1 | 2 | 3 | 4 | 5 |

(a)

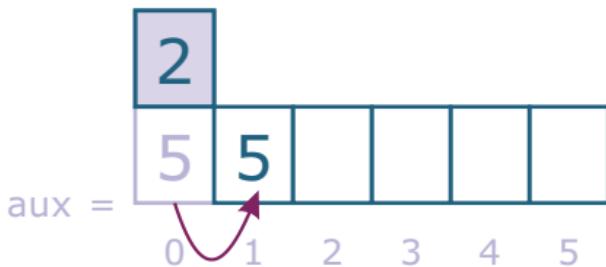
aux =

| | | | | | |
|---|---|---|---|---|---|
| 5 | | | | | |
| 0 | 1 | 2 | 3 | 4 | 5 |

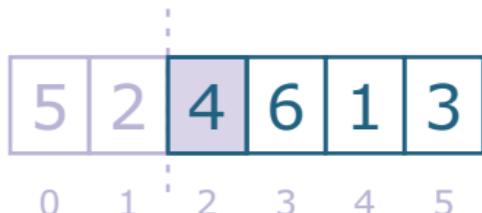
Dinâmica de Funcionamento



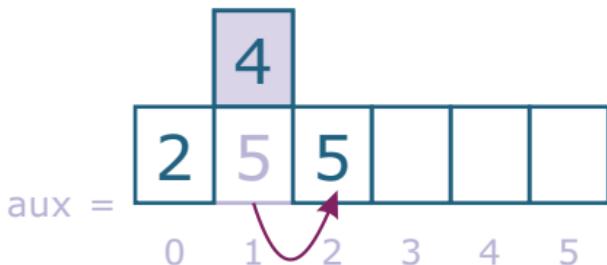
(b)



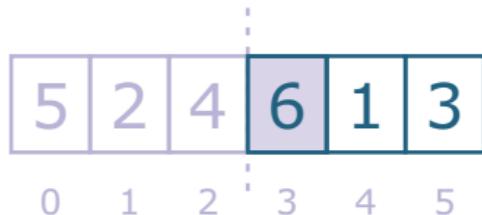
Dinâmica de Funcionamento



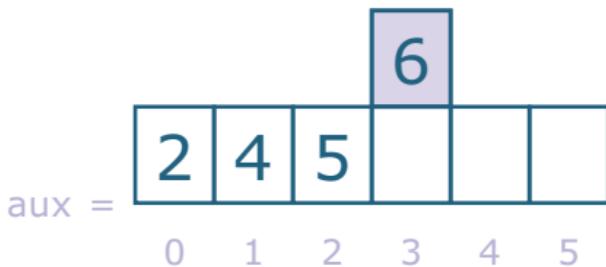
(c)



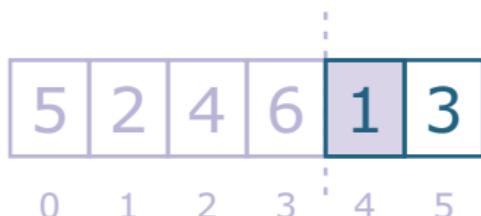
Dinâmica de Funcionamento



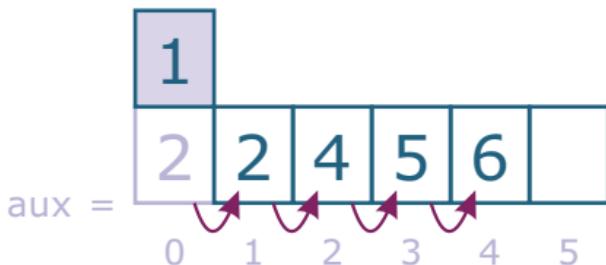
(d)



Dinâmica de Funcionamento



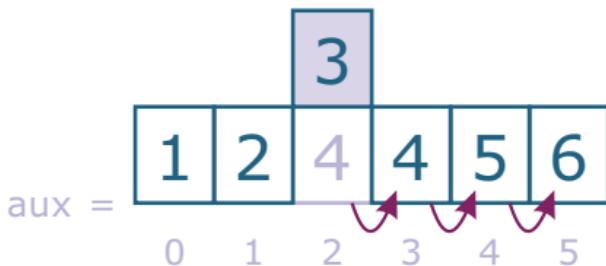
(e)



Dinâmica de Funcionamento



(f)



Implementação

Laço externo

Laço interno

Comparação e realocação dos demais elementos.

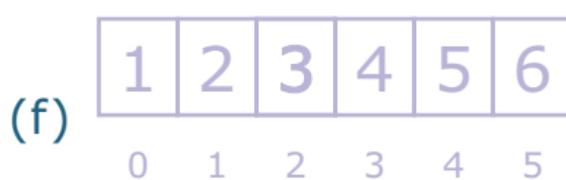
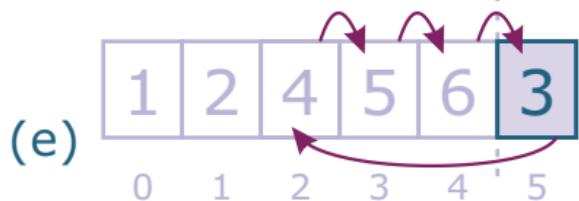
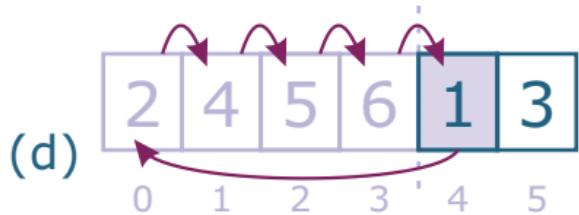
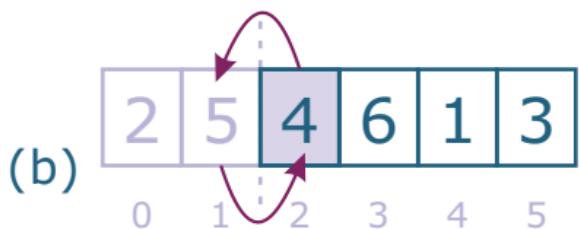
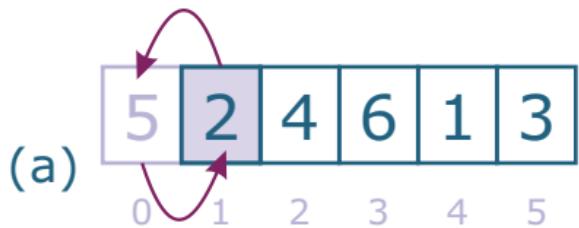
Inserção do elemento i da sequência original na ordem correta.

Insertion Sort

Lógica de Implementação sem Matriz Auxiliar

- Inicialmente, ordene os **dois primeiros** membros da matriz.
- Em seguida, **insira o terceiro membro** na sua posição ordenada com relação aos dois primeiros membros.
- Então, insira o **quarto** elemento na lista dos três elementos, também obedecendo a ordem.
- O processo continua até que todos os elementos tenham sido inseridos/ordenados.

Dinâmica de Funcionamento



Implementar

(1) Bubble Sort.

(2) Insertion Sort (sugere-se com o uso de matriz auxiliar).

Bubble Sort

Implementação Sugerida

```
int j, aux;
for(i = 0; i < TAM-1; i++)
{
    for (j = 0; j < (TAM - i-1); j++)
    {
        if (dados[j] > dados[j+1])
        {
            aux = dados[j];
            dados[j] = dados[j+1];
            dados[j+1] = aux;
        }
    }
}
```

Insertion Sort

Implementação Sugerida com Matriz Auxiliar

```
int novo[TAM], j, aux;
for (i = 0; i < TAM; i++)
{
    for (j = i; j > 0 && dados[i] < novo[j-1]; j--)
    {
        // aux = novo[j];
        novo[j] = novo[j-1];
        // novo[j-1] = aux;
    }
    novo[j] = dados[i];
}
```

Insertion Sort

Implementação Sugerida sem Matriz Auxiliar

```
int j, atual;
for (i = 1; i < TAM; i++)
{
    atual = dados[i];
    for (j = i - 1; j >= 0 && dados[j] > atual; j--)
    {
        dados[j+1] = dados[j];
    }
    dados[j+1] = atual;
}
```