Estácio

ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

ESTRUTURA DE DADOS

AULA 4

RICARDO EIJI KONDO, Me.

TÓPICOS

UNIDADE VI - Ordenação

6.1 - Introdução a Ordenação

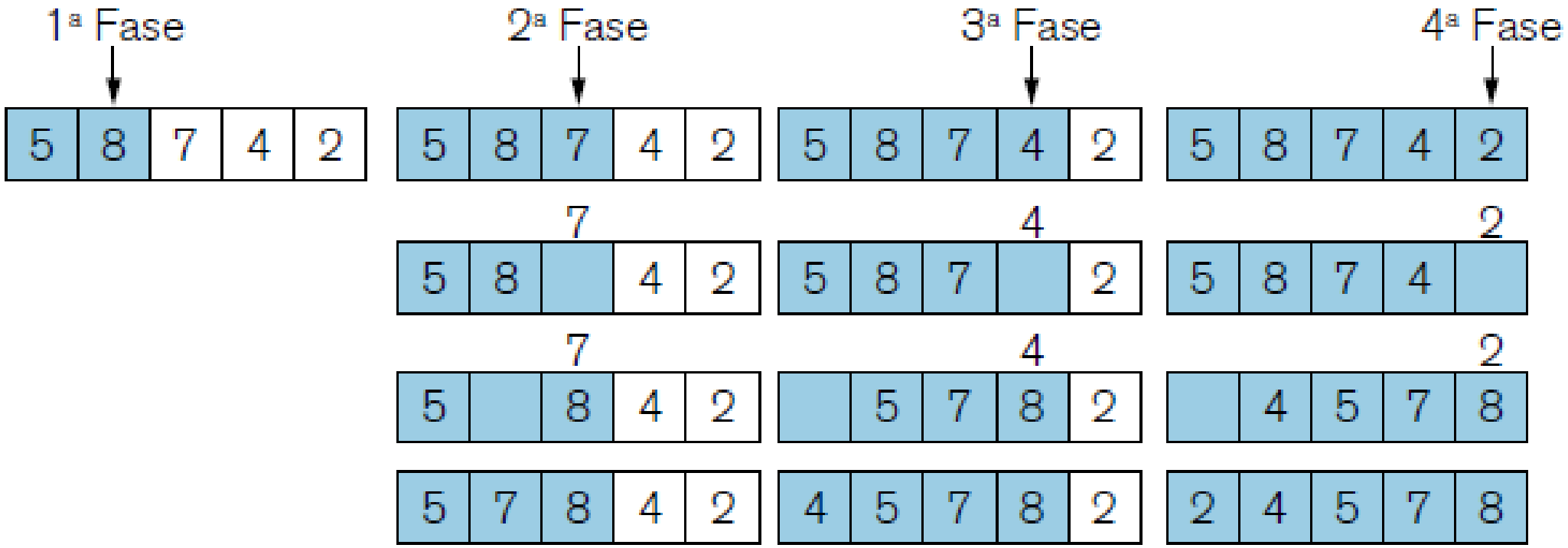
6.2 – Insertion sort, Selection sort, Bubble sort

ORDENAÇÃO

- Por que ordenar?
  - Facilidade na recuperação de um determinado elemento deste conjunto
- Podem-se adotar duas abordagens ao inserir um determinado elemento na lista
  - Respeitar a ordenação da estrutura;
  - Aplicar algum algoritmo de ordenação a um conjunto de dados já criado.

- **Métodos de ordenação**
  - Vários são os fatores que influência no desempenho de um algoritmo de ordenação, tais como: a quantidade de dados a serem ordenados, se todos os dados caberão na memória interna disponível, forma utilizada pelo algoritmo para ordenar os dados, etc.
  - Alguns métodos de ordenação mais importantes: Bubble Sort (ou ordenação por flutuação), Heap Sort (ou ordenação por heap), Insertion Sort (ou ordenação por inserção), Merge Sort (ou ordenação por mistura) e o Quicksort.

- **Insertion Sort (Inserção)**
  - Com base em um valor compara com as demais. Em seguida, insere o valor no local correto.

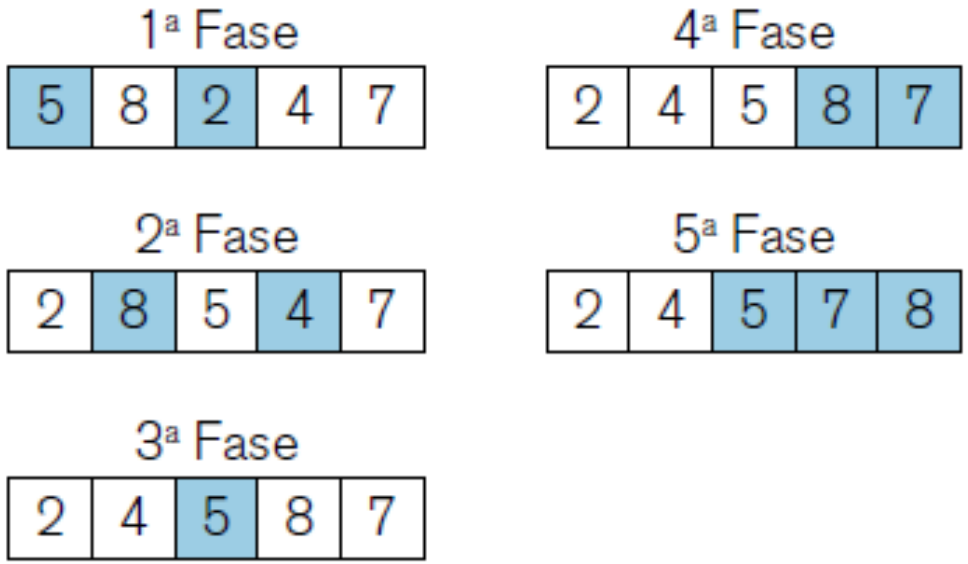


- **Insertion Sort (Inserção)**

```
 1  #include <iostream>
 2  using namespace std;
 3
 4  //função
 5  void insertionSort(int vetor[], int n)
 6  {
 7      int i, j, atual;
 8      for (i = 1; i < n; i++) {
 9          atual = vetor[i];
10          j = i-1;
11
12          while (j >= 0 && vetor[j] > atual)
13          {
14              vetor[j+1] = vetor[j];
15              j = j-1;
16          }
17          vetor[j+1] = atual;
18      }
19  }
```

```
21  //principal
22  int main()
23  {
24      int vetor[] = {12, 44, 23, 5, 1};
25
26      insertionSort(vetor, 5);
27      for (int i=0; i < 5; i++){
28          cout<<vetor[i]<<"\t";
29      }
30      return 0;
31  }
```

- **Selection Sort (Seleção)**
  - Percorre o vetor e coloca na primeira posição o menor valor. Em seguida, varre novamente o vetor, partindo da segunda posição. Encontrando o menor valor, este é colocado na segunda posição...

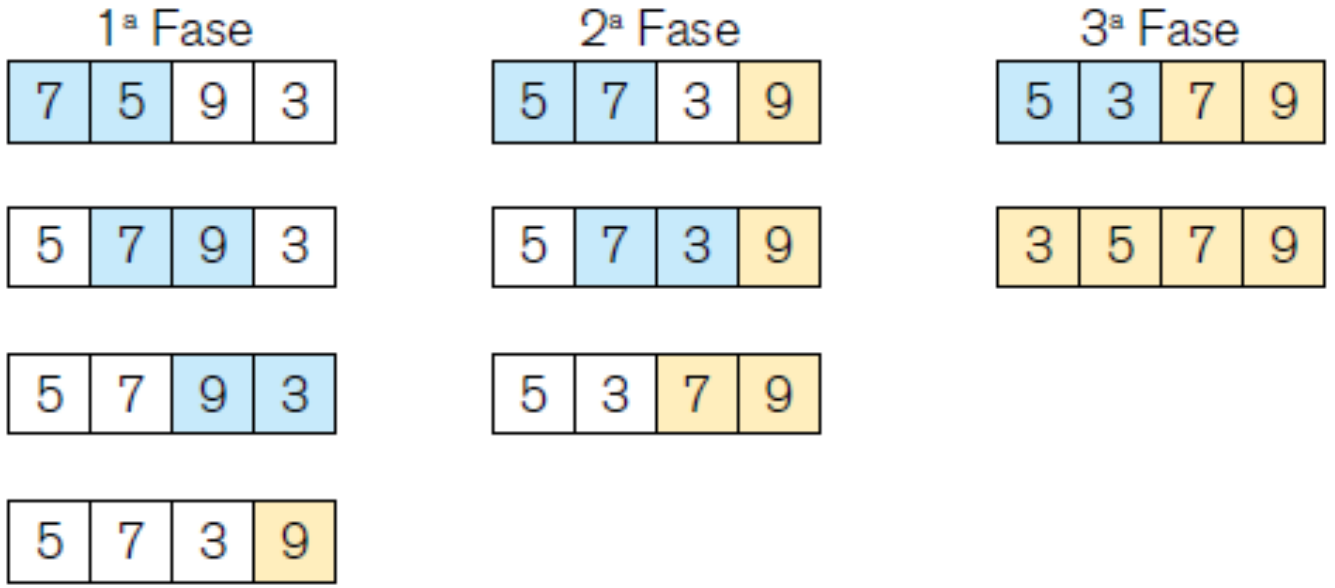


- **Selection Sort (Seleção)**

```
#include<iostream>
using namespace std;

void SelectionSort(int vetor[], int n){
    int i,j,indice,temp,minimo;

    for(i=0;i<n-1;i++)
    {
        minimo=vetor[i];
        indice=i;
        for(j=i+1;j<n;j++)
        {
            if(minimo>vetor[j])
            {
                minimo=vetor[j];
                indice=j;
            }
        }
        temp=vetor[i];
        vetor[i]=vetor[indice];
        vetor[indice]=temp;
    }
}
```

```
int main()
{
    int vetor[] = {12, 44, 23, 5, 1};

    SelectionSort(vetor, 5);
    for (int i=0; i < 5; i++){
        cout<<vetor[i]<<"\t";
    }
    return 0;
}
```

- **Bubble Sort (Flutuação por bolha)**
  - Consiste na troca de valores entre posições consecutivas.



- **Bubble Sort (Flutuação por bolha)**

```
#include<iostream>
using namespace std;

//função
void bubbleSort(int vetor[], int n)
{
  int i,j,temp;
    for(i=1;i<n;++i)
    {
        for(j=0;j<(n-i);++j)
        1{
            if(vetor[j]>vetor[j+1])
            {
                temp=vetor[j];
                vetor[j]=vetor[j+1];
                vetor[j+1]=temp;
            }
        }
    }
}
```

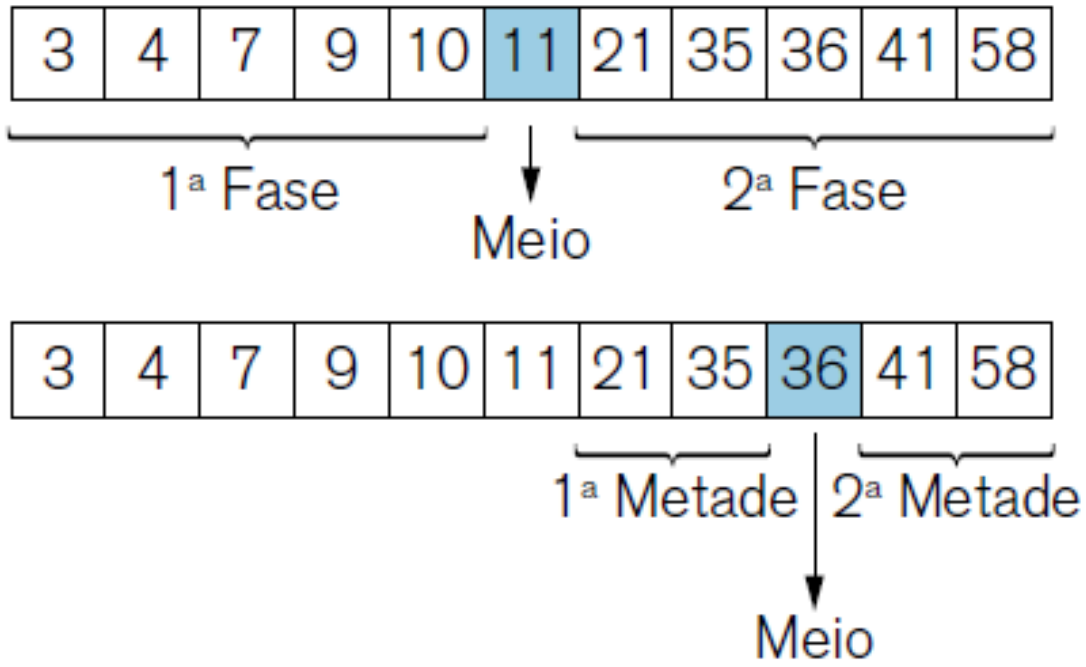
```
//principal
int main()
{
    int vetor[] = {12, 44, 0, 5, 1};

    bubbleSort(vetor, 5);
    for (int i=0; i < 5; i++){
        cout<<vetor[i]<<"\t";
    }
    return 0;
}
```

- **Pesquisa sequencial**
  - Percorre o vetor, comparando o elemento de interesse com cada elemento do vetor até que encontre o elemento procurado.

```
1  #include<iostream>
2  using namespace std;
3
4  //função
5  int pesquisaSequencial(int vetor[], int qtde, int valorBusca)
6  {
7      for (int i=0; i<qtde; i++)
8      {
9          if (vetor[i]==valorBusca)
10         {
11             return i;
12         }
13     }
14     return -1;
15 }
```

```
17  //principal
18 int main()
19 {
20     int vetor[] = {12, 44, 3, 5, 1};
21     cout<<pesquisaSequencial(vetor, 5, 12);
22     return 0;
23 }
```

- **Pesquisa binária**
  - Localizar o elemento central do vetor e compará-lo ao elemento procurado, para isso o vetor deve estar ordenado. Caso o elemento central for maior que o elemento procurado, então a próxima procura será na segunda parte do vetor.



- **Pesquisa binária**

```
#include<iostream>
using namespace std;
const int QTDE=5;

//procedimento ordenação
void insertionSort(int vetor[], int QTDE)
{
    int i, j, atual;
    for (i = 1; i < QTDE; i++) {
        atual = vetor[i];
        j = i-1;

        while (j >= 0 && vetor[j] > atual)
        {
            vetor[j+1] = vetor[j];
            j = j-1;
        }
        vetor[j+1] = atual;
    }
}
```

```
//função
int pesquisaBinaria(int vetor[], int QTDE, int valorBusca){

    int inf = 0;              //Limite inferior
    int sup = QTDE-1;    //Limite superior
    int meio;
        while (inf <= sup)
        {
            meio = (inf + sup)/2;
            if (valorBusca == vetor[meio])
                return meio;
            else if (valorBusca < vetor[meio])
                sup = meio-1;
            else
                inf = meio+1;
        }
        return -1; // não encontrado
}
```

- **Pesquisa binária**

```
42  //imprime
43  void print(int vetor[]){
44      for (int i=0; i < QTDE; i++){
45          cout<<vetor[i]<<"\t";
46      }
47      cout<<endl;
48  }
```

```
50   //principal
51  int main()
52  {
53      int vetor[] = {12, 99, 23, 5, 1};
54      int busca;
55      insertionSort(vetor, QTDE);
56      print(vetor);
57      cout<<"Digite o valor de busca"<<endl;
58      cin>>busca;
59      cout<<"A posicao do numero eh: "<<pesquisaBinaria(vetor,QTDE,busca);
60      return 0;
61  }
```

EXERCICIOS

Crie a estrutura a seguir e atribua os respectivos valores:

Id (int)	**Nome (string)**
5	Jose
7	Maria
2	Carlos
11	Ana
6	Bia

1) Ordene a lista pelo id utilizando o bubble sort e mostre na tela os ids e nomes
2) Utilize a pesquisa binária para localizar o id número 7 e mostre em qual posição ele se encontrar
3) Utilize a pesquisa binária para localizar a Ana e mostre em qual posição ela se encontrar