

Merge Sort

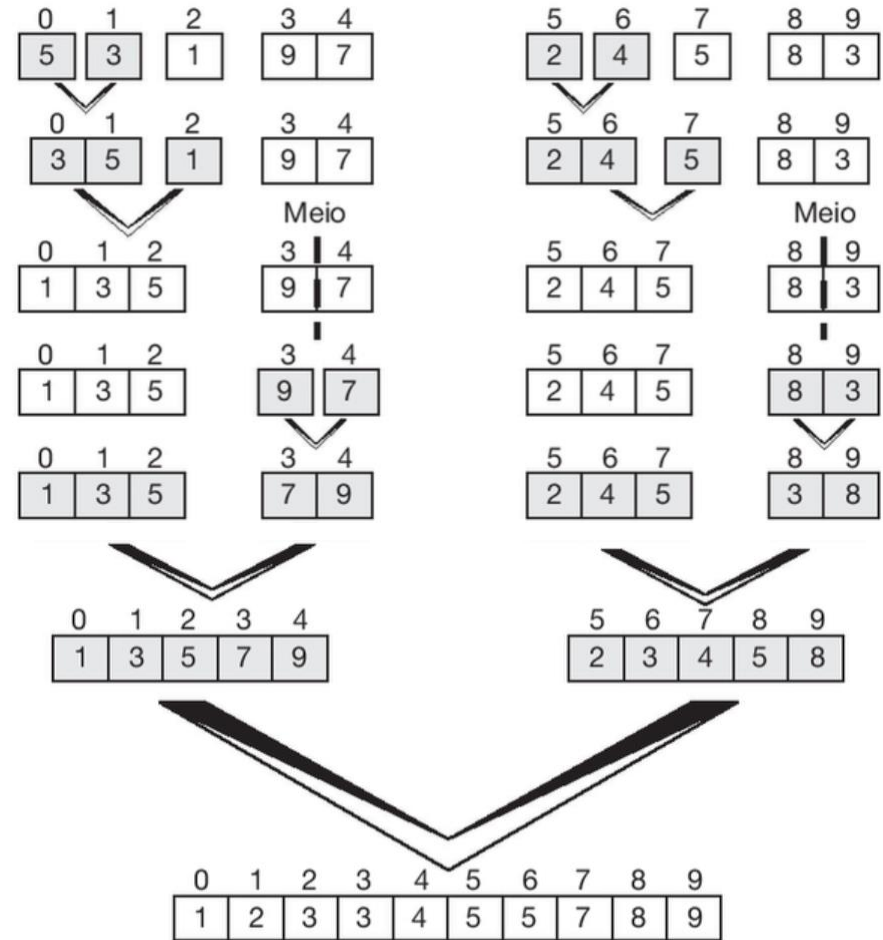
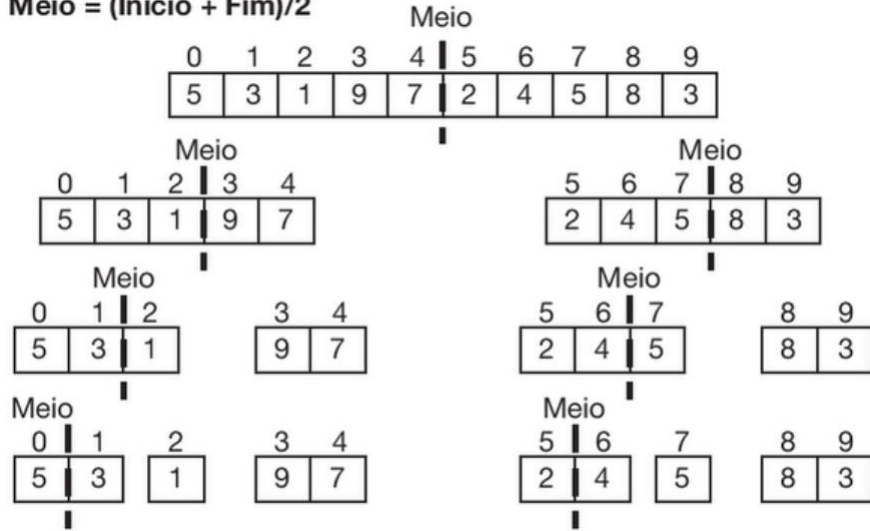
Murilo Dantas

Merge Sort

- Funcionamento
 - ▶ O vetor é dividido em vetores com a metade do tamanho original, de forma recursiva.
 - ▶ Isso ocorre até que o vetor fique com apenas um elemento e estes sejam ordenados e intercalados.

Merge Sort: exemplo

Meio = (Início + Fim)/2



Merge Sort: código

```
#include <iostream.h>
#include <conio.h>

void intercala (int X[],int inicio, int fim, int meio)
{
    int poslivre, inicio_vetor1, inicio_vetor2, i;
    int aux[10];
    inicio_vetor1 = inicio;
    inicio_vetor2 = meio + 1;
    poslivre = inicio;
    while (inicio_vetor1 <= meio && inicio_vetor2 <= fim)
    {
        if (X[inicio_vetor1] <= X[inicio_vetor2])
        {
            aux[poslivre] = X[inicio_vetor1];
            inicio_vetor1 = inicio_vetor1 + 1;
        }
        else
        {
            aux[poslivre] = X[inicio_vetor2];
            inicio_vetor2 = inicio_vetor2 + 1;
        }
        poslivre = poslivre + 1;
    }
}
```

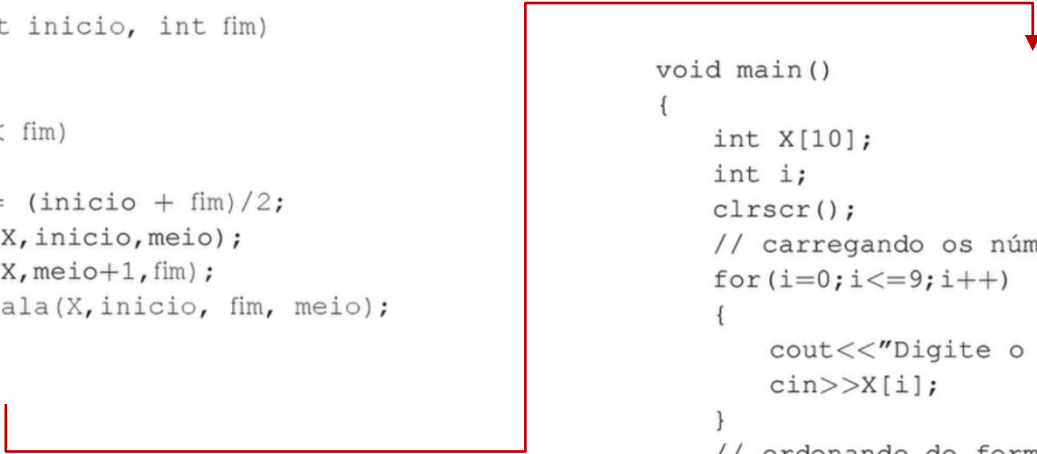
```
// se ainda existem números no primeiro vetor
// que não foram intercalados
for (i=inicio_vetor1;i<=meio;i++)
{
    aux[poslivre] = X[i];
    poslivre = poslivre + 1;
}

// se ainda existem números no segundo vetor
// que não foram intercalados
for (i=inicio_vetor2;i<=fim;i++)
{
    aux[poslivre] = X[i];
    poslivre = poslivre + 1;
}

// retorna os valores do vetor aux para o vetor X
for (i=inicio;i<=fim;i++)
{
    X[i] = aux[i];
}
}
```

Merge Sort: código

```
void merge (int X[], int inicio, int fim)
{
    int meio;
    if (inicio < fim)
    {
        meio = (inicio + fim)/2;
        merge(X, inicio, meio);
        merge(X, meio+1, fim);
        intercala(X, inicio, fim, meio);
    }
}
```



```
void main()
{
    int X[10];
    int i;
    clrscr();
    // carregando os números no vetor
    for(i=0; i<=9; i++)
    {
        cout<<"Digite o "<<i+1<<"º número: ";
        cin>>X[i];
    }
    // ordenando de forma crescente
    merge(X, 0, 9);
    // mostrando o vetor ordenado
    for(i=0; i<=9; i++)
    {
        cout<<"\n"<<i+1<<"º número: "<<X[i];
    }
    getch();
}
```

Perguntas?

Bibliografia da aula

- ASCENCIO, A. F. G.; ARAÚJO, G. S. Estrutura de dados. Algoritmos, análise da complexidade e implementação em Java e C/C++. Pearson. 2010.