

Classificação por Intercalação *MergeSort*

INF 01124 – Classificação e Pesquisa de Dados



MergeSort

- Dividir para Conquistar
 - **Dividir**: Divide a sequência de n elementos a ser classificada em duas subsequências de $n/2$ elementos
 - **Conquistar**: Classifica as duas subsequências recursivamente utilizando o *mergesort*
 - **Combinar**: Intercala as subsequências ordenadas
- A operação chave é a **intercalação** de duas sequências ordenadas
 - O processo de intercalação inicia com segmentos de comprimento 1, e prossegue até que toda a sequência esteja classificada

INF 01124 – Classificação e Pesquisa de Dados

Profa. Renata Galante



Intercalação de Arquivos

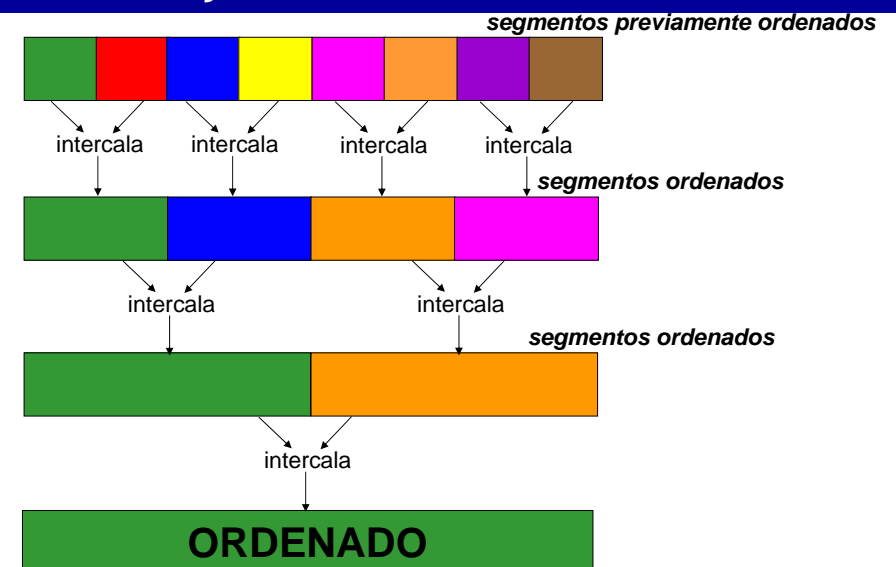
- o arquivo não cabe inteiramente na memória principal
- quando se deseja explorar recursos de paralelismo disponíveis no equipamento
- quando se deseja explorar o melhor desempenho de certos métodos de classificação, diante de uma menor quantidade de dados



INF 01124 – Classificação e Pesquisa de Dados

Profa. Renata Galante

Intercalação

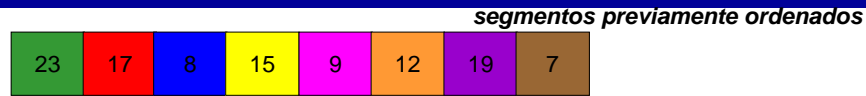


INF 01124 – Classificação e Pesquisa de Dados

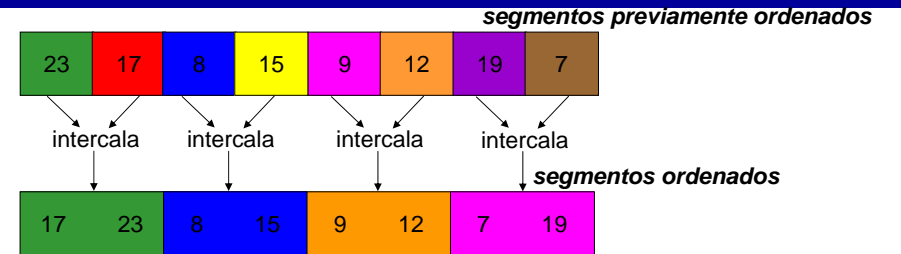
Profa. Renata Galante



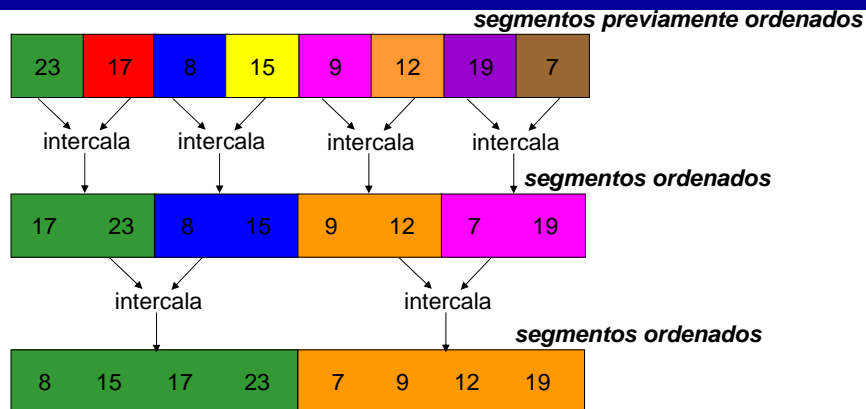
Intercalação Simples



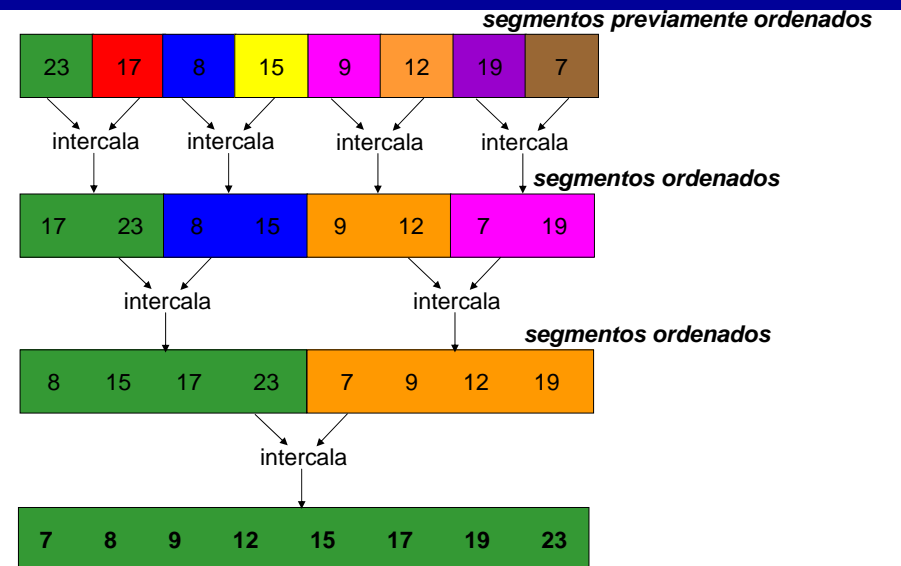
Intercalação Simples



Intercalação Simples



Intercalação Simples



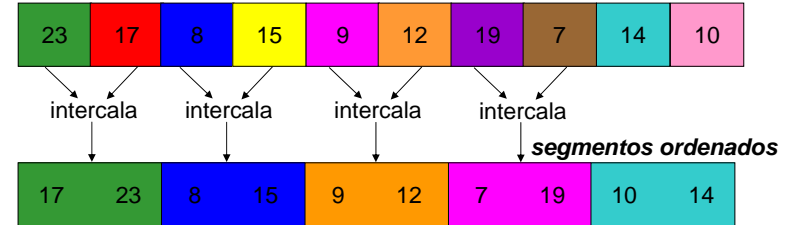
Intercalação Simples (NÃO potência de 2)

segmentos previamente ordenados



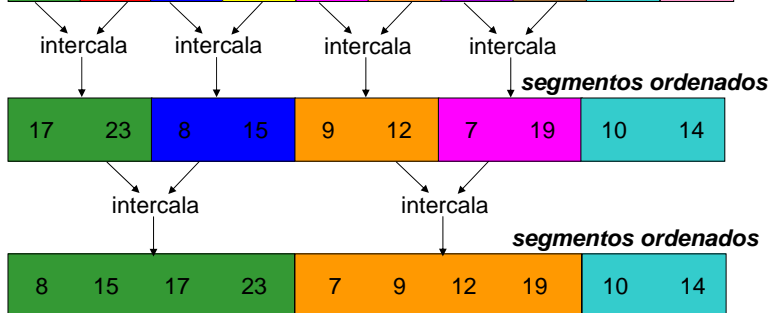
Intercalação Simples (NÃO potência de 2)

segmentos previamente ordenados



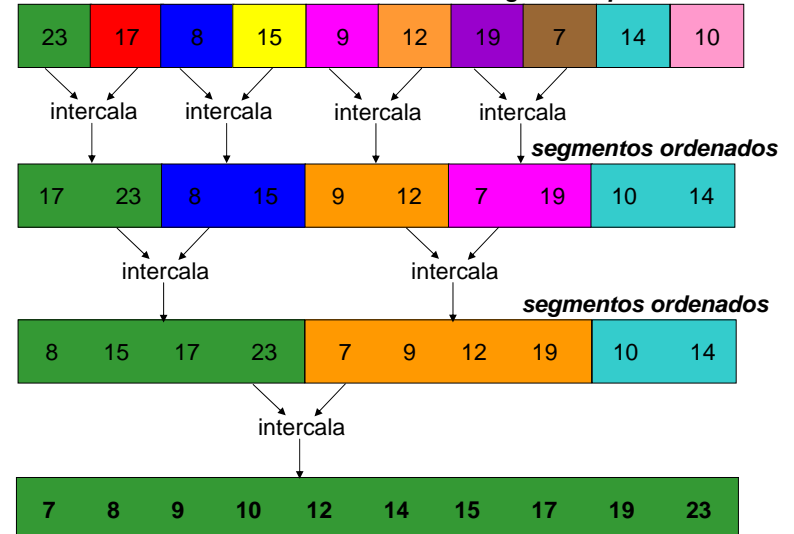
Intercalação Simples (NÃO potência de 2)

segmentos previamente ordenados



Intercalação Simples (NÃO potência de 2)

segmentos previamente ordenados



Função MergeSort (recursão)

rearranja o vetor $v[p..r-1]$

```
void ordenarMergeSort(int p, int r, int v[]){  
    if (p < r - 1){  
        int q = (p + r) / 2;  
  
        ordenarMergeSort(p, q, v);  
        ordenarMergeSort(q, r, v);  
        intercala(p, q, r, v);  
    }  
}
```

vetor

MergeSort : execução recursiva



Função MergeSort (iterativa)

```
void ordenarMergeSort (int n, int v[])  
{  
    int p, r;  
    int b = 1;  
    while (b < n) {  
        p = 0;  
        while (p + b < n) {  
            r = p + 2*b;  
            if (r > n) r = n;  
            intercala (p, p+b, r, v);  
            p = p + 2*b;  
        }  
        b = 2*b;  
    }  
}
```

Exercício

- Escrever uma função em C para fazer a intercalação de duas sequências ordenadas

Intercala

```
void intercala(int p, int q, int r, int v[]){
    int i, j, k, *w;
    w = malloc( (r - p) * sizeof( int) );
    i = p;
    j = q;
    k = 0;

    while (i < q && j < r){
        if (v[i] <= v[j])
            w[k++] = v[i++];
        else
            w[k++] = v[j++];
    }

    while (i < q) w[k++] = v[i++];
    while (j < r) w[k++] = v[j++];

    for (i = p; i < r; i++)
        v[i] = w[i - p];

    free(w);
}
```

Intercalação de Sequencias Naturais

- Utilizar ordenações parciais do vetor

