Ordenação por

Divisão e Conquista

Merge-sort

Ordenação Por Divisão e Conquista - Merge Sort

- Algoritmo de Ordenação por Divisão e Conquista, ou *Merge Sort*;
- Baseia-se na ideia de se dividir um vetor em vetores menores, até se obter vários pares de vetores unitários, o que simplifica a ordenação entre os esses pares;
- Algoritmo recursivo, utiliza-se da estratégia "Dividir para Conquistar" (*Divide and Conquer*).

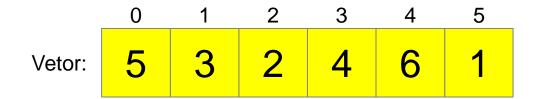
3/43

Estruturas de Dados e Análise de Algoritmos

Ordenação Por Divisão e Conquista - Merge Sort

- Sua complexidade no tempo é: O(n log, n);
- Sua complexidade no espaço é: O(n).

Ordenação Por Divisão e Conquista - Merge Sort



Vetor qualquer, supostamente desordenado!

Ordenação Por Divisão e Conquista - Merge Sort





O algoritmo ordenará o vetor crescentemente

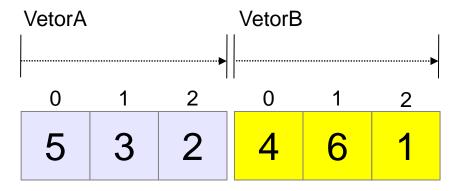
ttor: Prof. Robson Calvetti - Todos os direitos reservados @.

Ordenação Por Divisão e Conquista - Merge Sort



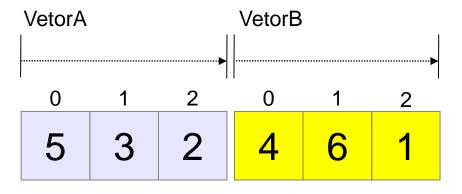
Etapa 1: Dividir o vetor em outros dois

Ordenação Por Divisão e Conquista - Merge Sort



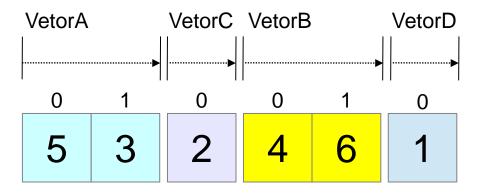
VetorA e VetorB

Ordenação Por Divisão e Conquista - Merge Sort



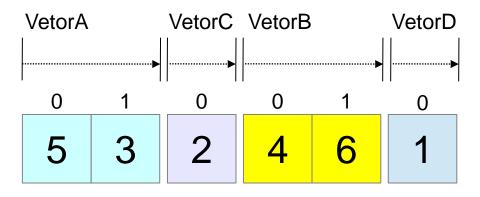
Etapa 2: Dividir cada vetor em outros dois

Ordenação Por Divisão e Conquista - Merge Sort



VetorA, VetorC, VetorB e VetorD

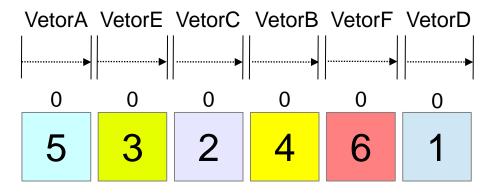
Ordenação Por Divisão e Conquista - Merge Sort



Etapa 3: Dividir cada vetor em outros dois

Prof. Calvetti 10/43

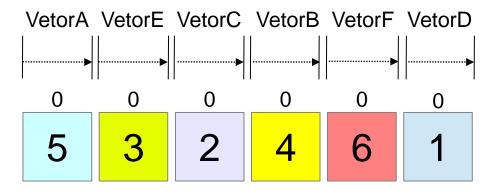
Ordenação Por Divisão e Conquista - Merge Sort



VetorA, VetorE, VetorC, VetorB, VetorF e VetorD

Prof. Calvetti 11/43

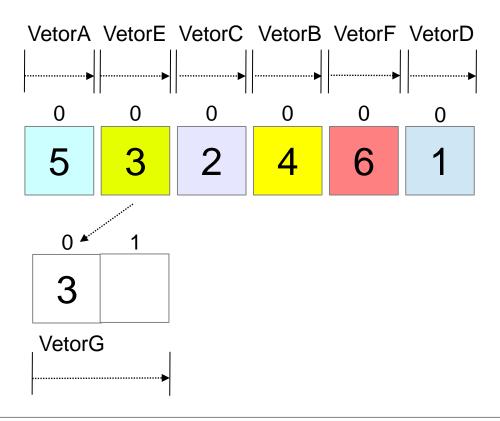
Ordenação Por Divisão e Conquista - Merge Sort



Etapa 4: Mesclar Ordenadamente os Pares

Prof. Calvetti 12/43

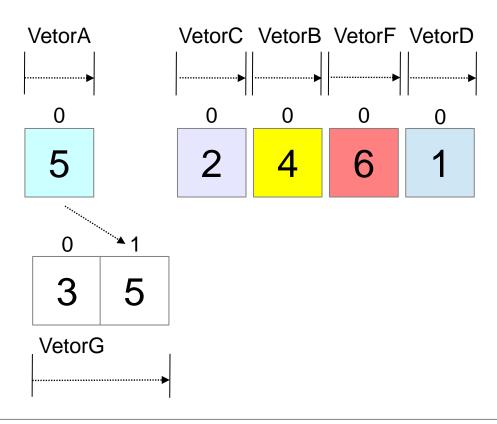
Ordenação Por Divisão e Conquista - Merge Sort



Cria VetorG para receber VetorA e VetorE

Prof. Calvetti 13/43

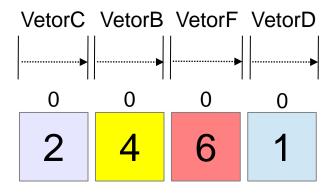
Ordenação Por Divisão e Conquista - Merge Sort

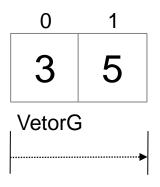


VetorG é mescla de VetorA e VetorE

Prof. Calvetti 14/43

Ordenação Por Divisão e Conquista - Merge Sort

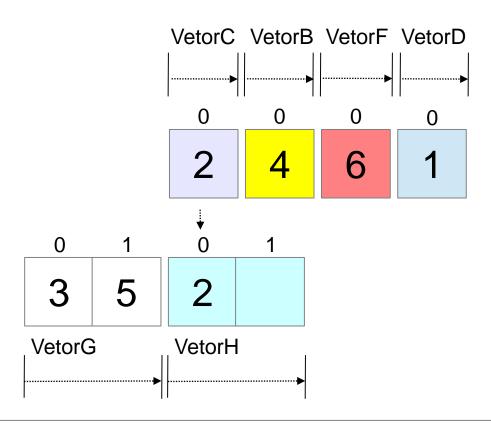




VetorG Ordenado!

Prof. Calvetti 15/43

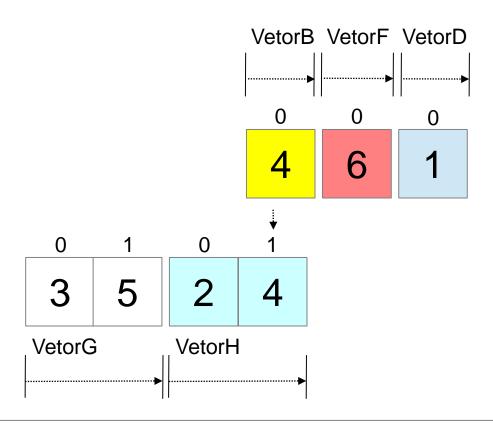
Ordenação Por Divisão e Conquista - Merge Sort



Cria VetorH para receber VetorC e VetorB

Prof. Calvetti 16/43

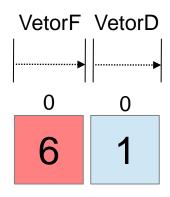
Ordenação Por Divisão e Conquista - Merge Sort

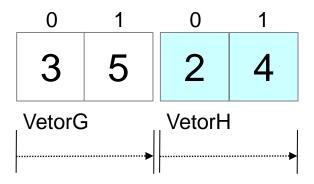


VetorH é mescla de VetorC e VetorB

Prof. Calvetti 17/43

Ordenação Por Divisão e Conquista - Merge Sort

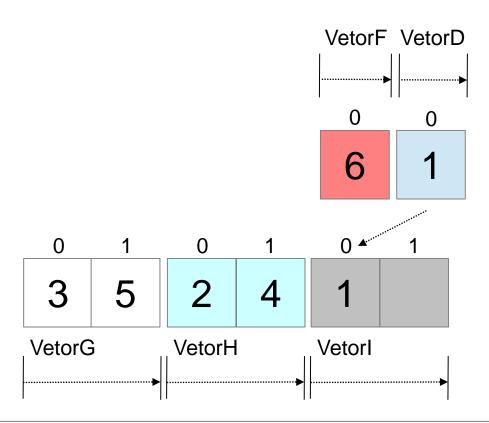




VetorH Ordenado!

Prof. Calvetti 18/43

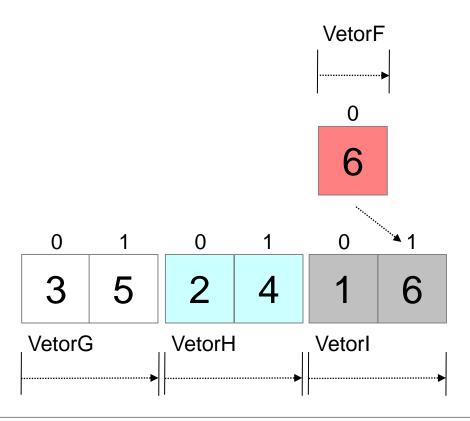
Ordenação Por Divisão e Conquista - Merge Sort



Cria VetorI para receber VetorF e VetorD

Prof. Calvetti 19/43

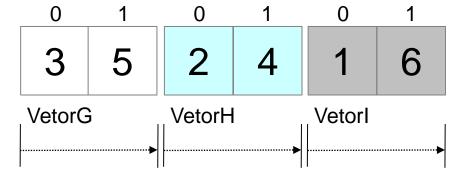
Ordenação Por Divisão e Conquista - Merge Sort



Vetorl é mescla de VetorF e VetorD

Prof. Calvetti 20/43

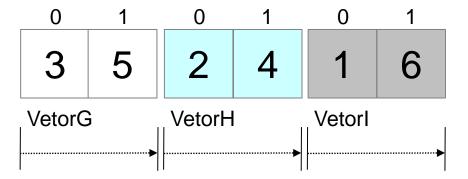
Ordenação Por Divisão e Conquista - Merge Sort



Vetorl Ordenado!

Prof. Calvetti 21/43

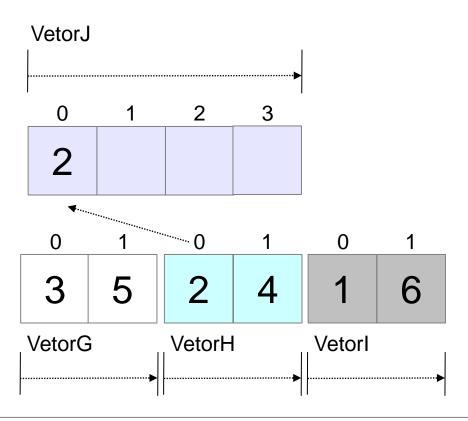
Ordenação Por Divisão e Conquista - Merge Sort



Etapa 5: Mesclar Ordenadamente os Pares

Prof. Calvetti 22/43

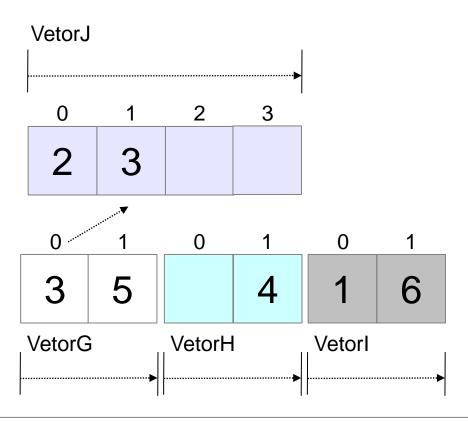
Ordenação Por Divisão e Conquista - Merge Sort



Cria VetorJ para receber VetorG e VetorH

Prof. Calvetti 23/43

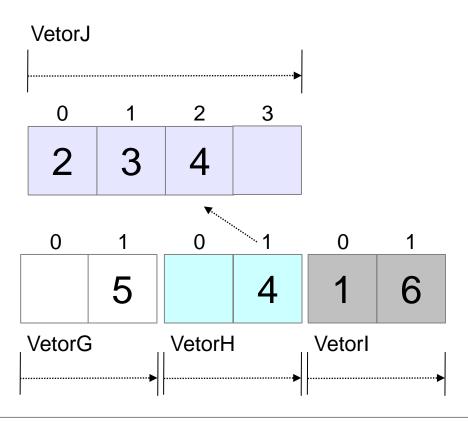
Ordenação Por Divisão e Conquista - Merge Sort



VetorJ é mescla de VetorG e VetorH

Prof. Calvetti 24/43

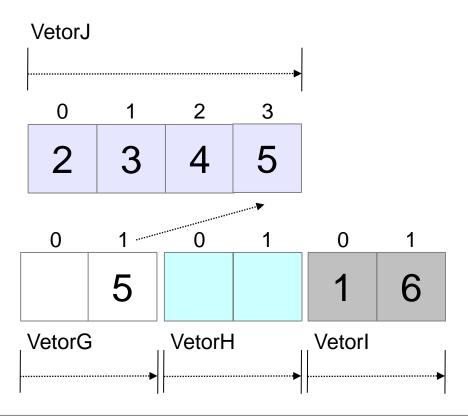
Ordenação Por Divisão e Conquista - Merge Sort



VetorJ é mescla de VetorG e VetorH

Prof. Calvetti 25/43

Ordenação Por Divisão e Conquista - Merge Sort

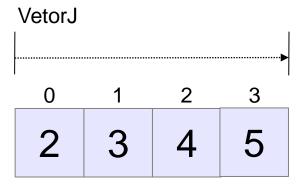


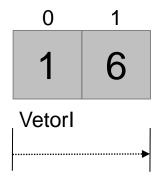
VetorJ é mescla de VetorG e VetorH

Prof. Calvetti 26/43

Estruturas de Dados e Análise de Algoritmos

Ordenação Por Divisão e Conquista - Merge Sort

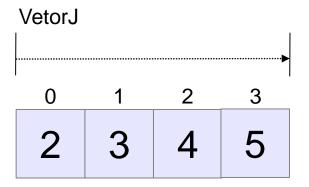


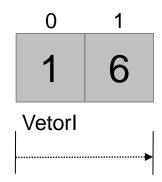


VetorJ Ordenado!

Prof. Calvetti 27/43

Ordenação Por Divisão e Conquista - Merge Sort

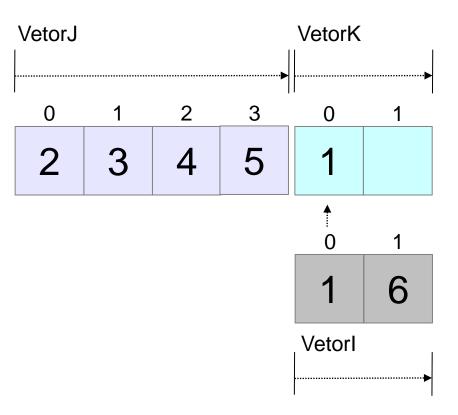




Etapa 6: Mesclar Ordenadamente os Pares

Prof. Calvetti 28/43

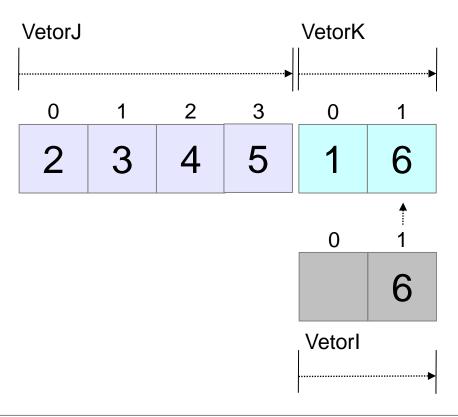
Ordenação Por Divisão e Conquista - Merge Sort



Cria VetorK para receber VetorI

Prof. Calvetti 29/43

Ordenação Por Divisão e Conquista - Merge Sort

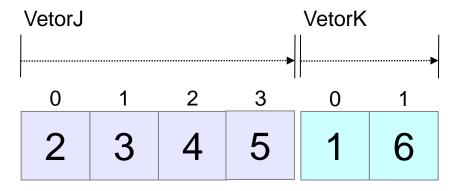


VetorK é o VetorI

Prof. Calvetti 30/43

Estruturas de Dados e Análise de Algoritmos

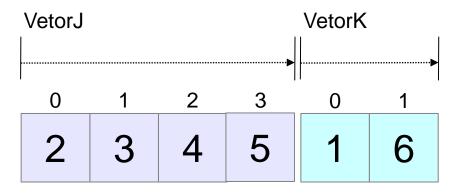
Ordenação Por Divisão e Conquista - Merge Sort



VetorK Ordenado!

Prof. Calvetti 31/43

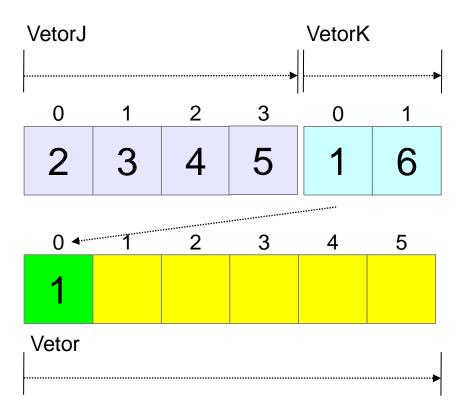
Ordenação Por Divisão e Conquista - Merge Sort



Etapa 7: Mesclar Ordenadamente os Pares

Prof. Calvetti 32/43

Ordenação Por Divisão e Conquista - Merge Sort

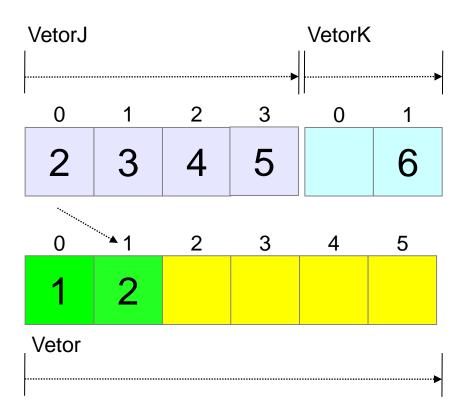


Usa Vetor para receber VetorJ e VetorK

Prof. Calvetti 33/43

Estruturas de Dados e Análise de Algoritmos

Ordenação Por Divisão e Conquista - Merge Sort

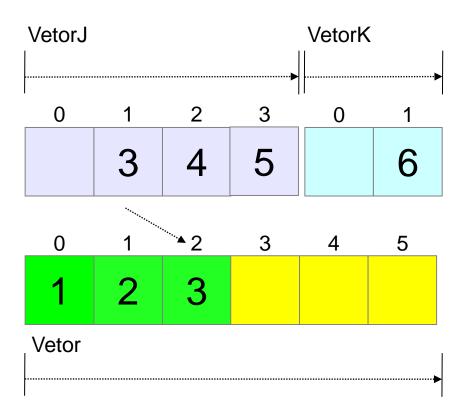


Vetor é mescla de VetorJ e VetorK

Prof. Calvetti 34/43

Estruturas de Dados e Análise de Algoritmos

Ordenação Por Divisão e Conquista - Merge Sort

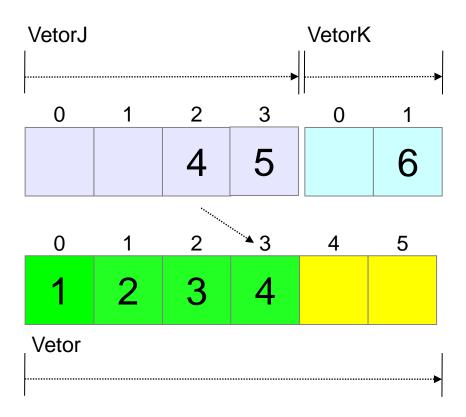


Vetor é mescla de VetorJ e VetorK

Prof. Calvetti 35/43

Estruturas de Dados e Análise de Algoritmos

Ordenação Por Divisão e Conquista - Merge Sort

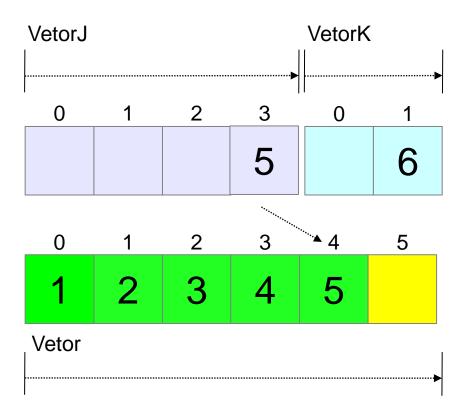


Vetor é mescla de VetorJ e VetorK

Prof. Calvetti 36/43

Estruturas de Dados e Análise de Algoritmos

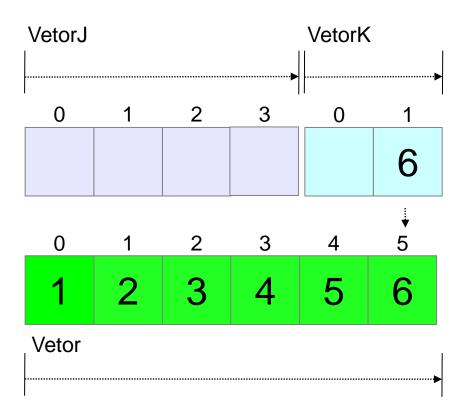
Ordenação Por Divisão e Conquista - Merge Sort



Vetor é mescla de VetorJ e VetorK

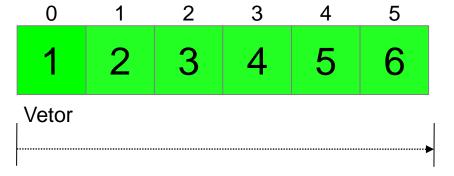
Prof. Calvetti 37/43

Ordenação Por Divisão e Conquista - Merge Sort



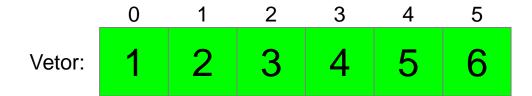
Vetor é mescla de VetorJ e VetorK

Prof. Calvetti 38/43



Vetor Ordenado!

Ordenação Por Divisão e Conquista - Merge Sort



Vetor Original Ordenado!

Prof. Calvetti 40/43

Ordenação Por Divisão e Conquista - Merge Sort

```
public static void mergeSort(int v[])
    if(v.length > 1)
        int t1 = v.length/2;
        int t2 = v.length-t1;
        int v1[] = new int[t1];
        int v2[] = new int[t2];
        for(int i = 0; i < t1; i++)
            v1[i] = v[i];
        for (int i = t1; i < (t1+t2); i++)
            v2[i-t1] = v[i];
        mergeSort (v1);
        mergeSort (v2);
        merge(v, v1, v2);
```

Prof. Calvetti 41/43

Ordenação Por Divisão e Conquista - Merge Sort

```
public static void merge(int w[], int w1[], int w2[])
{ int i = 0, j = 0, k = 0;
  while (w1.length != j && w2.length != k)
     if(w1[j] >= w2[k])
        w[i] = w1[j];
            i++;
            j++;
        else
           w[i] = w2[k];
            i++;
            k++;
    while (w1.length != j)
     w[i] = w1[i];
        i++;
        j++;
    while (w2.length != k)
     w[i] = w2[k];
        i++;
        k++;
```

Prof. Calvetti 42/43

Ordenação Por Divisão e Conquista - Merge Sort

FIM

Prof. Calvetti 43/43