UniEVANGÉLICA
UNIVERSIDADE EVANGÉLICA DE GOIÁS

Estrutura de Dados

Métodos de Ordenação

Merge Sort

Prof.(a): Me. William P. Santos Júnior

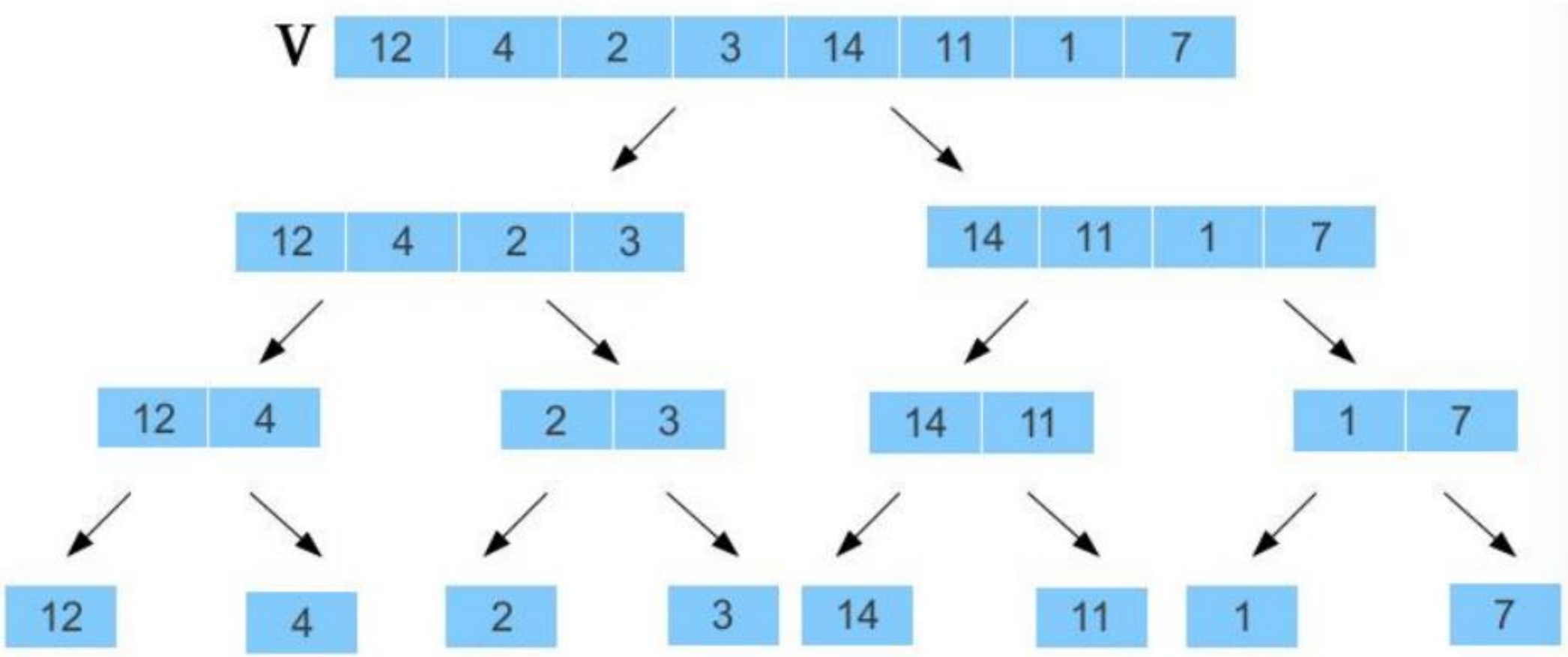
Merge Sort – Dividir e Conquistar

- Devemos resolver um problema P de tamanho N
- **Dividir:** Temos que quebrar P em sub-problemas menores
- Resolvemos os Sub-problemas usando recursividade
- **Conquistar:** Fazemos a união das soluções dos Sub-problemas para obter a solução final que é o problema P

Merge Sort – Ordenação por Intercalação

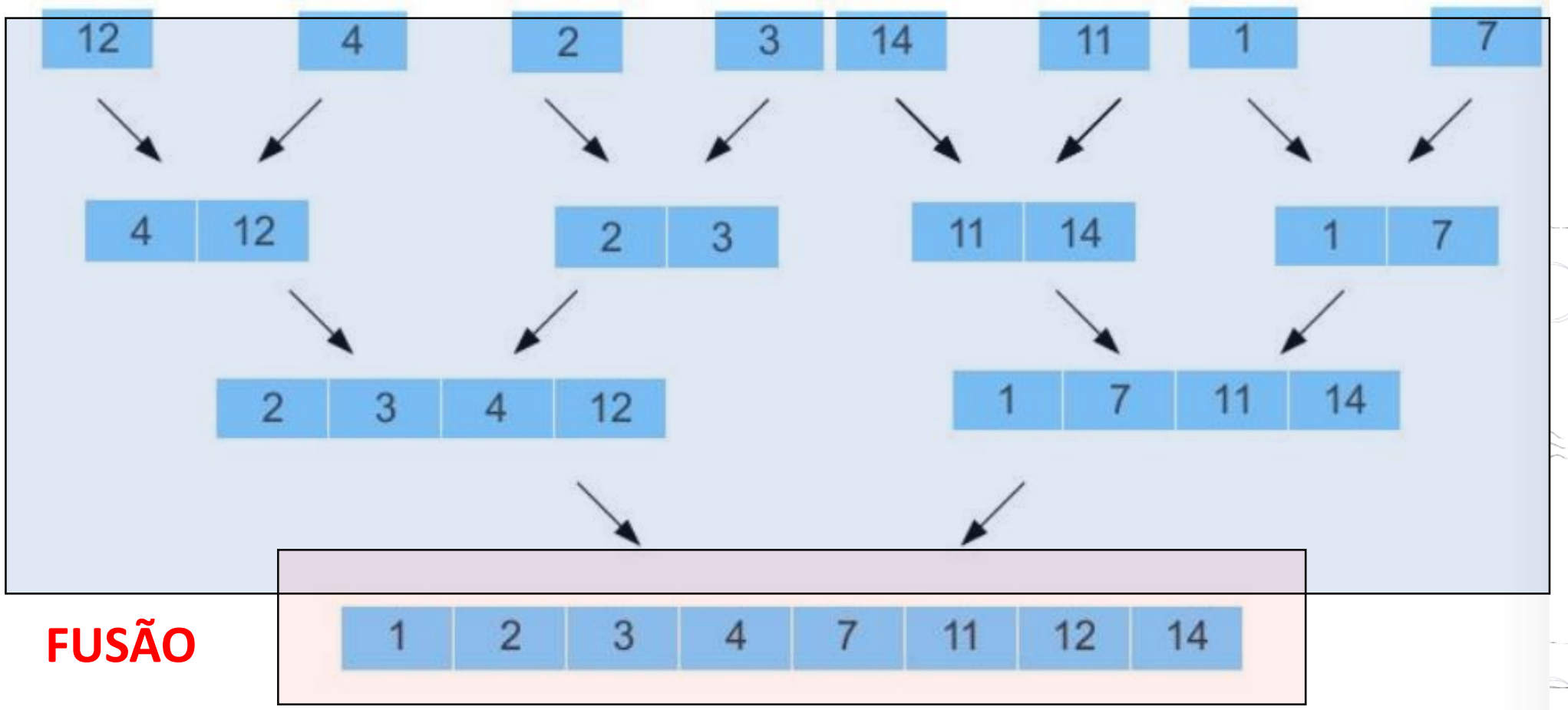
- Baseado na técnica de dividir e conquistar
- Para esse caso devemos ordenar um vetor de tamanho ***N***
  - ***Dividir:*** Dividimos o vetor de tamanho ***N*** em dois sub-vetores com aproximadamente o mesmo tamanho. (n / 2).
  - Resolvemos o problema de ordenação de forma recursiva para estes dois sub-vetores.
  - ***Conquistar:*** Com os dois sub-vetores ordenados, devemos reconstruir o vetor ordenado de tamanho ***N. Merge***

Merge Sort



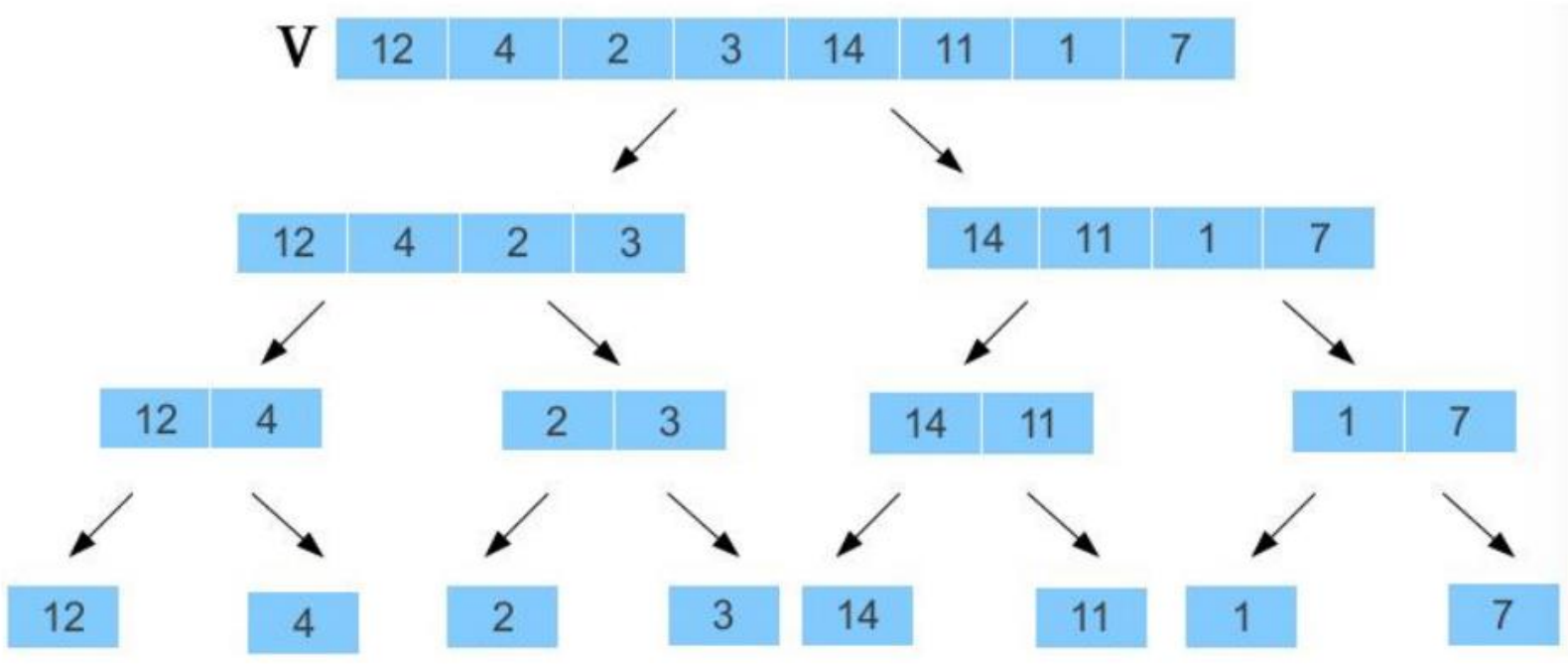
Merge Sort

DIVIDE O VETOR

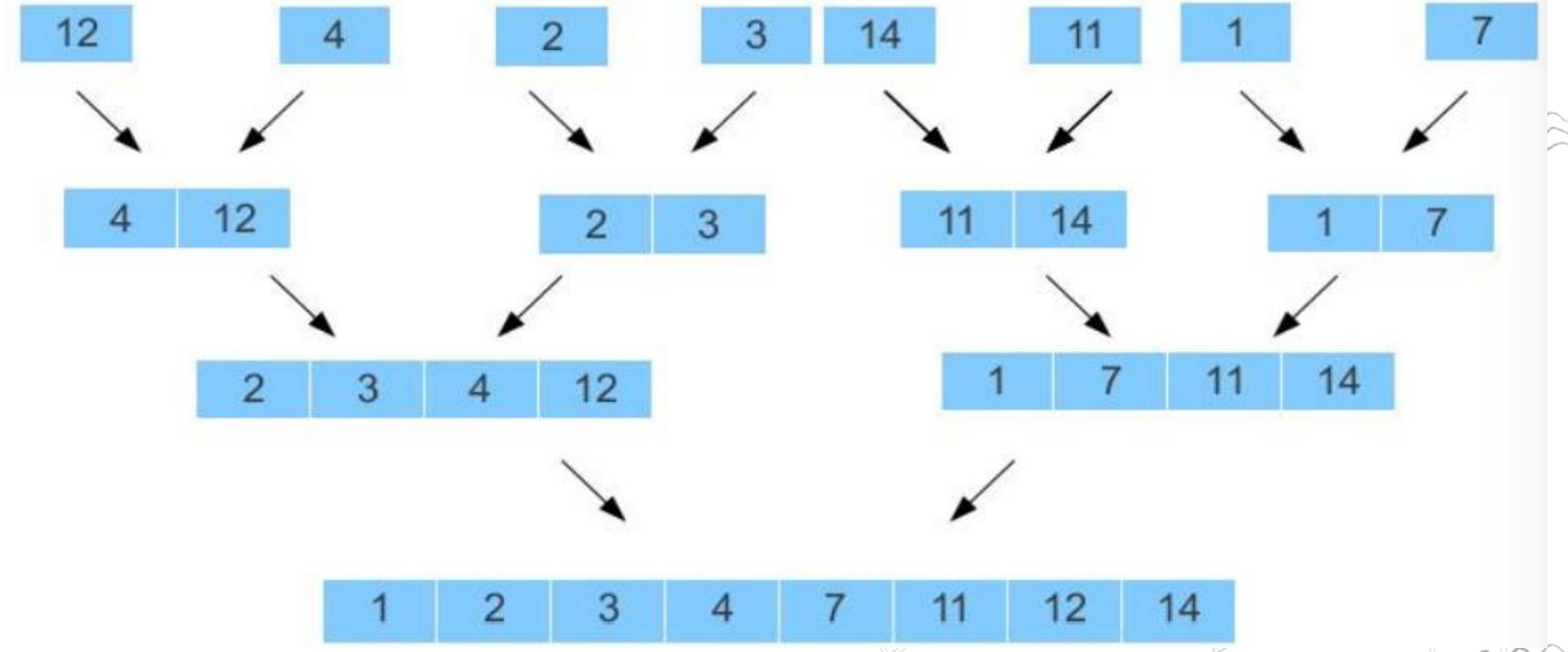


FUSÃO

1ª Parte

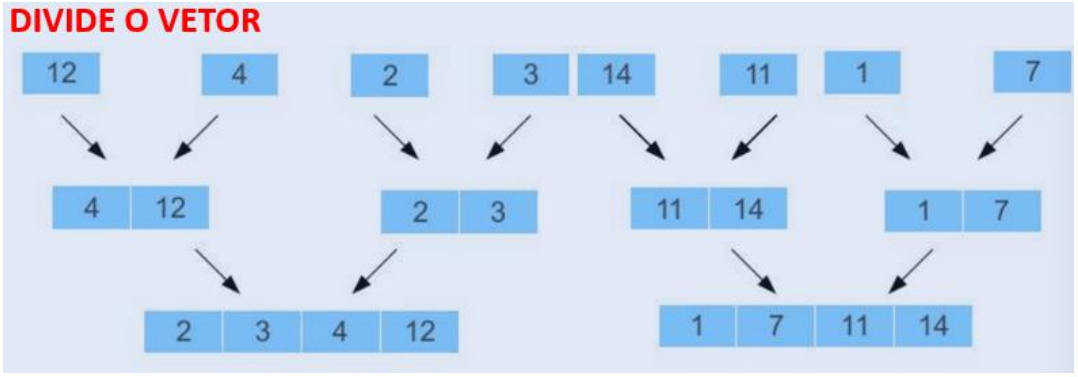


2ª Parte



Merge Sort

DIVIDE O VETOR

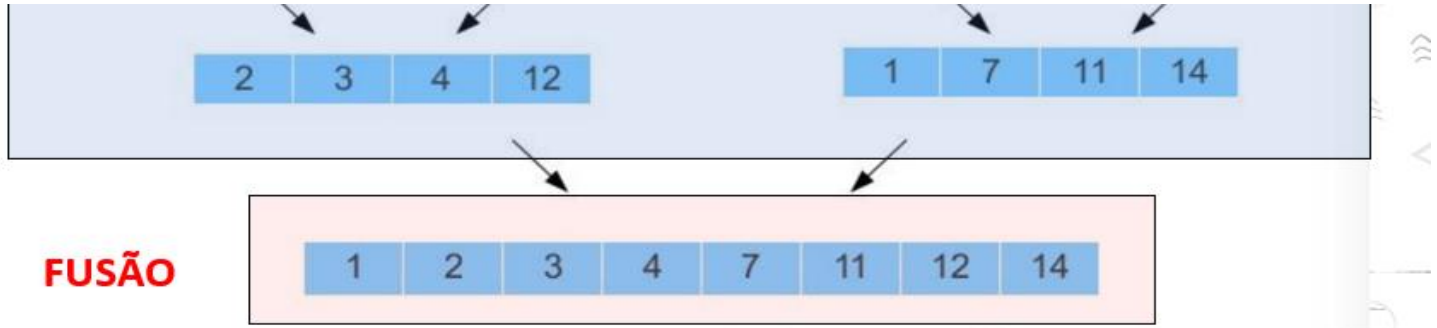



```
void merge(int vet[], int inicio, int meio, int fim){
    int tLadoE = meio - inicio + 1; // tamanho do subVetor da esquerda
    int tLadoD = fim - meio; // tamanho do subVetor da Direita
    int esq[tLadoE]; // subVetor auxiliar para o lado esquerdo
    int dir[tLadoD]; // subVetor auxiliar para o lado direito
    int idxEsq = 0;  // indice para o subVetor auxiliar esquerdo
    int idxDir = 0;  // indice para o subVetor auxiliar direito
    int i, j, k;

    // faz a cópia dos elementos do subVetor do lado esquerdo para o vetor aux
    for(i = 0; i < tLadoE; i++){
        esq[i] = vet[inicio + i];
    }
    // faz a cópia dos elemento so subVetor do lado direito para o aux
    for(j = 0; j < tLadoD; j++){
        dir[j] = vet[meio +1 + j];
    }
    // fazendo a intercalação dos vetores
    for(k = inicio; k <= fim; k++){
        if(idxEsq < tLadoE){
            if(idxDir < tLadoD){
                if(esq[idxEsq] < dir[idxDir]){
                    vet[k] = esq[idxEsq++];
                }
                else{
                    vet[k] = dir[idxDir++];
                }
            }
            else{
                vet[k] = esq[idxEsq++];
            }
        }
        else{
            vet[k] = dir[idxDir++];
        }
    }
}
```

Merge Sort

FUSÃO




```
void mergeSort(int vet[], int inicio, int fim){
    if(inicio < fim){
        int meio = (inicio + fim)/2; // acha o meio do vetor
        mergeSort(vet, inicio, meio);// ordena o subVetor da esquerda
        mergeSort(vet, meio+1, fim); // ordena o subVetor da direita
        merge(vet,inicio, meio, fim); // faz a  junção dos vetores da esquerda/direita
    }
}
```

UniEVANGÉLICA

UNIVERSIDADE EVANGÉLICA DE GOIÁS