

## Estruturas de Dados II: ed202\_ex02 mergesort

1 – Alterar o algoritmo de *mergesort* para, ao final, mostrar o número de trocas realizadas.

2 – Executar o algoritmo com o vetor usado na aula para intercalar os vetores:  $v1 = \{28, 33, 66, 70\}$  e  $v2 = \{39, 60, 78, 95\}$ .

3 – Um algoritmo de ordenação é estável se preserva ordem relativa de itens com valores idênticos.

2.1 - O insertsort é estável? Exemplifique. Se não é estável, como podemos modificá-lo para ser estável?

2.2 - O mergesort é estável? Exemplifique. Se não é estável, como podemos modificá-lo para ser estável?

4 – Modifique o mergesort para fazer ordenação decrescente.

5 – Escreva e analise uma versão iterativa do mergesort.

6 – Qual o melhor caso do mergesort? Qual seu custo neste caso?

7 – Escreva uma versão do mergesort para ordenar listas encadeadas. Sua solução não deve alocar novas células na memória.

7.1 – Escreva uma versão iterativa.

7.2 – Escreva uma versão recursiva.

8 – Qual a complexidade computacional do mergesort? Explique

9 – Discuta a seguinte implementação do algoritmo Mergesort:

```
void mergesort1( int p, int r, int v[]) {
    if (p < r-1) {
        int q = (p + r) / 2;
        mergesort1( p, q, v);
        mergesort1( q, r, v);
        intercala( p, q+1, r, v);
    }
}
```

10 - Discuta a seguinte implementação do algoritmo Mergesort:

```
void mergesort2( int p, int r, int v[]) {
    if (p < r) {
        int q = (p + r) / 2;
        mergesort2( p, q, v);
        mergesort2( q, r, v);
        intercala( p, q, r, v);
    }
}
```

11 - Discuta a seguinte implementação do algoritmo Mergesort:

```
void mergesort3( int p, int r, int v[]) {
    if (p < r-1) {
        int q = (p + r - 1) / 2;
        mergesort3( p, q, v);
        mergesort3( q, r, v);
        intercala( p, q, r, v);
    }
}
```

12 - Discuta a implementação do algoritmo Mergesort que está abaixo. Depois, tente mais uma versão, com  $(p+r+1)/2$  no lugar de  $(p+r)/2$ .

```
void mergesort4( int p, int r, int v[]) {
    if (p < r-1) {
        int q = (p + r) / 2;
        mergesort4( p, q-1, v);
        mergesort4( q-1, r, v);
        intercala( p, q-1, r, v);
    }
}
```

13 - Critique a seguinte versão do Mergesort:

```
void mergesort5( int p, int r, int v[]) {
    if (p < r-1) {
        q = r - 1;
        mergesort5( p, q, v);
        intercala( p, q, r, v);
    }
}
```