

Universidade Tecnológica Federal do Paraná (UTFPR)
Departamento Acadêmico de Informática (DAINF)
Estrutura de Dados I
Professor: Rodrigo Minetto
Lista de exercícios (algoritmos básicos de ordenação)

Exercícios (seleção): necessário entregar **TODOS** (moodle)!

Exercício 1) Codifique e execute os algoritmos bubble-sort e selection-sort para ordenar um quantidade de 10, 100, 1.000, 10.000, 100.000 e 200.000 elementos em ordem **aleatória**. Indique na tabela abaixo o tempo para cada uma das entradas acima. **Não inclua** no tempo de execução a impressão dos elementos na tela (comente a função **print** para estes testes de performance).

Algoritmo	10	100	1.000	10.000	100.000	200.000
Bubble-Sort						
Selection-Sort						

Para auxiliar implemente a funcionalidade que falta conforme indicado nos programas **bubble.c** e **selection.c** (em **arquivos.zip**).

Exercício 2) Implemente uma versão recursiva do algoritmo **selection-sort**. Não é necessário eliminar todos os iteradores, basta escolher **um loop** para transformar em recursão — para isso, observe qual deles é menos acoplado ao código. Utilize o seguinte protótipo para a sua função:

```
void selection_sort_recursive (int *A, int n);
```

Não altere o protótipo acima, ou seja, não adicione outras variáveis como argumentos de entrada na função para facilitar o projeto da recursão.

Exercício 3) Implemente uma versão recursiva do algoritmo **bubble-sort**. Não é necessário eliminar todos os iteradores, basta escolher **um loop** para transformar em recursão — para isso, observe qual deles é menos acoplado ao código. Utilize o seguinte protótipo para a sua função:

```
void bubble_sort_recursive (int *A, int n);
```

Não altere o protótipo acima, ou seja, não adicione outras variáveis como argumentos de entrada na função para facilitar o projeto da recursão.

Exercício 4) Implemente uma versão alternativa do algoritmo **bubble-sort** cuja ideia é deslocar o maior elemento da iteração atual até o final do array e no retorno, deslocar o menor elemento até o início do array. Suponha por exemplo o seguinte array $A = \{5, 4, 3, 2, 1\}$, então o processo de ordenação ocorre da seguinte forma:

I) 5 4 3 2 1 entrada/input
1) 4 3 2 1 5 deslocando maior
2) 1 4 3 2 5 deslocando menor
3) 1 3 2 4 5 deslocando maior
4) 1 2 3 4 5 deslocando menor

Utilize o seguinte protótipo para a sua função:

```
void bubble_sort_bidirecional (int *A, int n);
```