

## Lista de Exercícios 6 (Métodos de Ordenação: Quick Sort)

### Observações:

- Crie uma pasta *Lista 6* no seu repositório de códigos (GitHub, repl.it, Drive) para disponibilizar os seus programas.
- Nos formulários de submissão das atividades submeta apenas o link deste repositório a fim de facilitar as entregas!

Faça uma cópia do seu programa desenvolvido para a Lista 5, que ordena números inteiros aleatórios e acrescente a seguinte funcionalidade:

1. Altere o arquivo `com112_main.c` para incluir no menu a opções 5:
  1. Bubble Sort
  2. Selection Sort
  3. Insertion Sort
  4. Merge Sort
  - 5. Quick Sort**
  6. Relatório
  7. Sair
2. Assim como as opções 1, 2, 3 e 4 do menu, quando o usuário selecionar a opção **5. Quick Sort** o programa deve chamar uma função implementada no arquivo `com112_sort.c` para ordenação dos números aleatórios pelo método *Quick Sort*.
3. Quando o usuário selecionar a opção **6. Relatório**, o programa deve executar todos os métodos de ordenação (agora incluindo o *Quick Sort*) sobre os dados obtidos do arquivo de entrada (`com112_entrada.txt`), imprimir na tela as informações estatísticas de desempenho de cada método e gravar tais dados no arquivo `com112_relatorio.txt`.
4. Para testar se os dados estatísticos de desempenho gerados pelo seu programa estão corretos, teste o programa com  $n = 10000$  e compare os dados obtidos com os dados do livro texto <sup>1</sup>.

**OBS:** os dados estatísticos obtidos pelo seu programa devem ser proporcionais aos mencionados no livro, nem sempre serão iguais pois depende da forma como o algoritmo foi implementado e dos números aleatórios gerados para o vetor de entrada.

---

<sup>1</sup>ASCÊNCIO, A. F. G.; ARAÚJO, Graziela S. *Estrutura de Dados: Algoritmos, Análise de Complexidade, Implementações em Java e C/C++*. 2010.