



**Disciplina Programação (CK0226)**  
**Tarefa de Laboratório FINAL**  
**Comparação do Tempo de Busca entre ABB, Vetor e Lista**

Prof. Miguel Franklin

**OBJETIVOS:**

Equipes de até 2 (dois) alunos

- Experimentação de eficiência de busca.
- Implementação de operação de remoção em estrutura de dados Árvore Binária de Busca (ABB).
- Prática de divisão de projeto em diversos arquivos-fonte.

**ENUNCIADO:**

1. Utilizar como base a lista duplamente encadeada de trabalhos anteriores e fazê-la armazenar apenas uma chave inteira e um valor associado do tipo ponto flutuante, e criar uma função de busca de valor passando a chave inteira.
2. Criar uma aplicação que extrai  $N$  tuplas compostas por chaves (inteiras) distintas e valores ponto flutuante pseudo-aleatórios quaisquer, armazenando-as em uma ABB e em uma Lista Duplamente Encadeada, na ordem em que esses valores são “sorteados”.
3. Criar uma função para percorrer a ABB na ordem crescente, armazenando os valores inteiros das chaves em um vetor de inteiros de  $N$  de posições, que será naturalmente ordenado.
4. Implementar uma função **recursiva** de busca na ABB, retornando apenas se a chave existe ou não na árvore.
5. Implementar uma função de busca binária<sup>1</sup> **recursiva** no vetor ordenado, retornando apenas se a chave existe ou não no vetor.
6. Implementar uma função **iterativa** de busca sequencial no mesmo vetor, retornando apenas se a chave existe ou não no vetor.
7. Medir o tempo de busca de  $K$  buscas de chaves aleatórias em cada uma das quatro modalidades de busca: busca em ABB, busca binária em vetor, busca sequencial em vetor e busca sequencial em Lista Duplamente Encadeada.
8. Exibir, no final, apenas os tempos de busca de cada uma das modalidades de busca.
9. Deve-se ajustar os valores de  $N$  e  $K$  para que o tempo de busca na ABB fique em torno de 2 segundos. Os valores de  $N$  e  $K$  devem ser definidos através de macros (`#define`) no `main.c`.
10. Cada estrutura de dados deverá ser implementada em arquivos fonte (.c e .h) distintos, assim como um arquivo fonte específico da aplicação (do teste de tempo – `main.c`), além de um `Makefile` que deverá construir o projeto.

*A entrega (upload) deverá ser realizada através do Google Classroom.*

---

<sup>1</sup> [https://www.ime.usp.br/~pf/analise\\_de\\_algoritmos/aulas/binarysearch.html](https://www.ime.usp.br/~pf/analise_de_algoritmos/aulas/binarysearch.html)