



### Instruções para entrega do roteiro:

- Entregue o roteiro apenas no formato *.pdf* com o nome ***Y\_roteiroX.pdf***, onde **X é o número do roteiro** e **Y é o número da sua matrícula**. Não serão aceitos outros formatos.
- Inclua nome e matrícula, e mantenha a resolução dos exercícios **ordenada e legível**.
- Códigos completos (com `int main`), compiláveis e executáveis, quando aplicável.  
**Para cada um, apresente uma imagem da tela de saída do seu programa.**
- Após a data de entrega, a nota da entrega é 0.
- Em caso de dúvidas, procurem os monitores. Haverá um monitor após as aulas de laboratório para tirar dúvidas sobre a lista.

## Roteiro 12

### Pesquisa e Hashing

Data máxima de entrega: 09/12/2023  
(Entrega: pelo SIGAA, na sua turma de laboratório.)

## 1 Algoritmos de Pesquisa

- 1.1 Reimplemente os algoritmos de busca sequencial e busca binária e teste-os comparando os valores finais de número de comparações e tempo de execução.
- 1.2 Em relação ao algoritmo de busca binária, quais alterações devem ser realizadas para que a busca seja feita em um vetor ordenado de forma decrescente? Aplique as mudanças nas versões recursiva e iterativa do método e teste o seu código.
- 1.3 Utilizando o código feito na aula prática para a ordenação de um vetor de struct, adapte esse código para que a struct seja de um aluno com os campos (nome, matrícula e 3 notas). Adapte o algoritmo de ordenação e busca binária, para que a busca seja realizada de 2 formas diferentes. A primeira seria de encontrar um aluno pelo nome, e a outra de encontrar pela matrícula. Teste o seu programa.

## 2 Tabela Hash

- 2.1 Reimplemente o TAD: Tabela Hash visto inicialmente sem o tratamento de colisões, insira a 3 diferentes funções de hashing (Divisão, Multiplicação e Dobra) e teste o espalhamento criado com cada uma delas e também a busca por elementos.
- 2.2 Dessa vez, altere o TAD anterior, para incluir o tratamento de colisões por **endereçamento aberto**, ou seja, escolha uma das funções de hashing e depois aplique as 3 diferentes formas de sondagem. Teste novamente a execução e também a busca por elementos.

- 2.3 Por fim, reimplente o TAD: Tabela Hash que faz uso do tratamento de colisões por **encadeamento separado**, dessa forma, escolha uma das funções de hashing e no caso de colisões, agora cada posição da tabela deve incluir o elemento em uma lista encadeada. Teste novamente a execução e também a busca por elementos.

Link do roteiro (read-only) L<sup>A</sup>T<sub>E</sub>X: <https://pt.overleaf.com/read/txzmvdtyyvvd#a4cd55>