

# Relatório de desempenho de algoritmos de buscas

**Autor:** Thiago de Oliveira Cordeiro

**Disciplina:** Estrutura de Dados Básicas I

**Professor:** Selan Rodrigues dos Santos

## 1. Contextualização

Como exposto durante as aulas, os **problemas de buscas** englobam os problemas cujas respostas podem ser strings arbitrárias. Ou seja, podemos definir como uma pesquisa em algum contêiner com o fito de constatar a presença e localizar um determinado elemento ou então descobrir a ausência do mesmo. E, por se tratar de um problema bastante comum na Ciência da Computação, é de importância a utilização de algoritmos eficientes que solucionem a problemática.

Entre os algoritmos utilizados para solucionar problemas de buscas, temos a **busca linear** e a **busca binária**. Na primeira, percorremos todo os elementos do contêiner e paramos quando encontramos o elemento, quando chegamos ao final da estrutura que contém os dados e não o achamos ou quando existe alguma informação extra sobre os dados que permite deduzir que o elemento procurado não está entre os demais, além disso, nesse algoritmo, também temos a garantia que teremos uma resposta em tempo finito e previsível.

Temos também a busca binária, onde começamos o algoritmo verificando se o intervalo de busca tem ao menos um elemento. Caso não tenha, pode-se ser feita alguma ação, indicando a ausência de elementos. Se tiver, procuramos o elemento localizado no meio do intervalo de pesquisa, nomeando-o de meio. Caso ele seja igual ao valor buscado, podemos retornar o índice desse elemento. Mas se o buscado é menor que o valor do meio, mudamos nossa área de pesquisa, reduzindo o intervalo de busca, indo do primeiro elemento até o valor antes do meio e calculando o novo meio, repetindo o processo de análise. Já se o alvo for maior do que o meio, mudamos nossa área de pesquisa, reduzindo o intervalo de busca, indo do elemento depois do meio até o último valor e calculando o novo meio, repetindo o processo de análise.

Seguimos os passos acima até constatar a presença do elemento buscado ou a sua ausência. Vale ressaltar que, para a busca binária e linear é necessário que não haja a repetição de elementos e para a busca binária, em particular, é necessário que os dados estejam ordenados.

## 2. Pesquisa

Para a realização da pesquisa, lançou-se mão do uso da análise de tempo de execução, onde, basicamente, busca-se compreender o comportamento de um componente de software, uma função, ou todo um software, fazendo análises de dados coletados durante a execução do software ou de parte do software.

E o presente trabalho analisou dois tipos de algoritmos de busca: o linear e o binário. Destaca-se que a comparação da busca linear e a binária foi realizada usando o recurso de iteração da linguagem de programação C++. Não só esse recurso foi usado do C++, mas também todos os algoritmos foram construídos no C++. Ademais, foi criado também um algoritmo de busca binária usando recursão, com o fito de comparar o desempenho com a busca binária iterativa. E então, analisou-se o tempo de execução de cada um dos algoritmos. Avaliando pelo pior caso, onde o elemento buscado não se encontra dentro do contêiner.

A mensuração de tempo foi feita usando a biblioteca Chrono Library do C++ e para os testes, foram criados vetores de diversos tamanhos.

## 3. Resultados

### 3.1 Busca Linear e Busca Binária

De acordo com os valores obtidos na pesquisa e anexados em um gráfico, pode-se perceber que a busca binária é realmente um algoritmo de busca mais eficiente que a busca linear, haja vista que o seu tempo de execução é bem menor que o algoritmo de busca linear no pior caso.

Segue abaixo o gráfico obtido com a pesquisa:

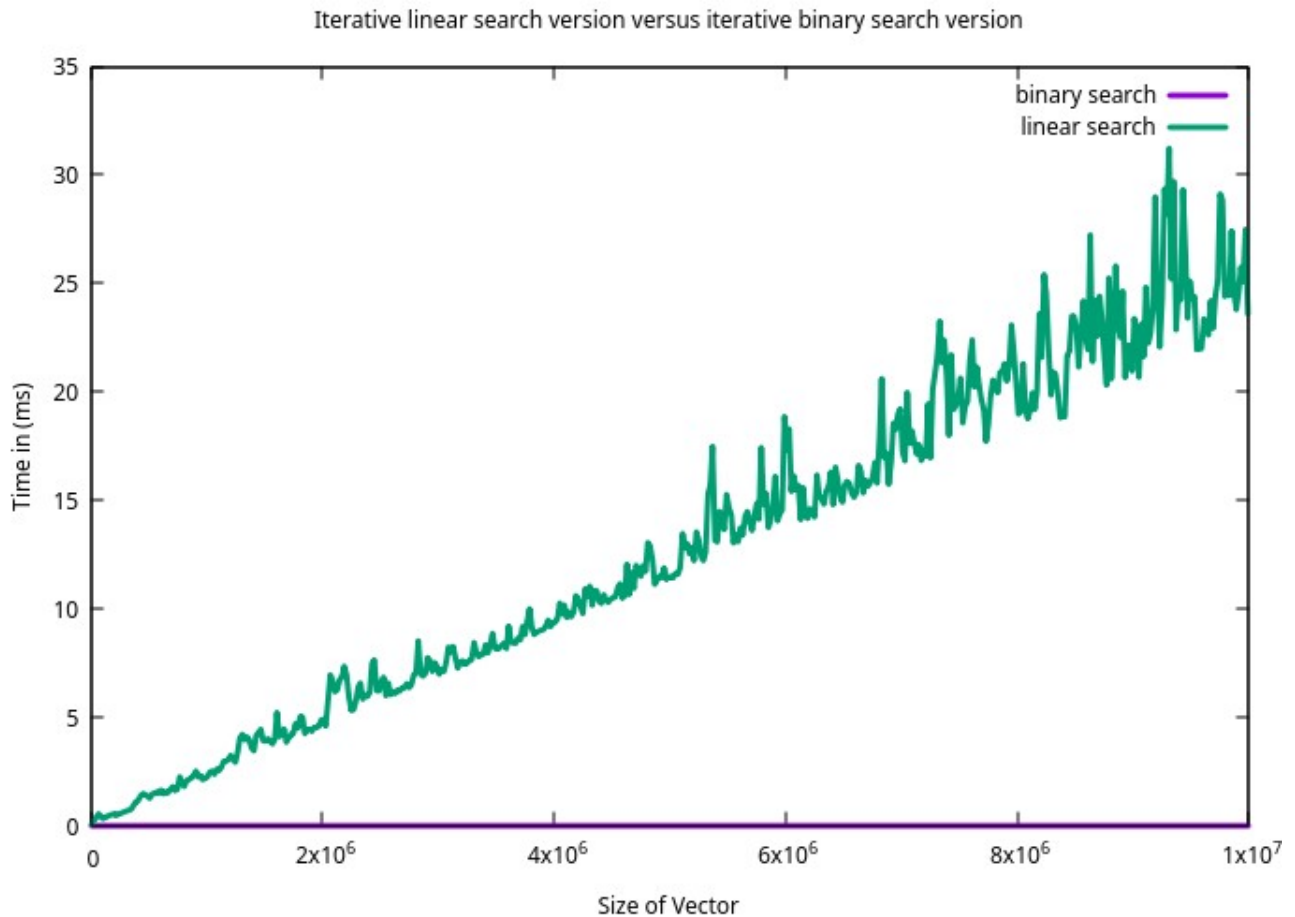


Figura 1: gráfico de comparação de desempenho da busca linear e binária

### 3.2 Busca Linear e Busca Binária

Segundo os dados obtidos nessa pesquisa, pode-se perceber que, de acordo com o que está exposto no gráfico não há diferença grande entre os dois algoritmos: o de busca binária usando recursão e o de busca binária usando iteração. Mas, em uma análise ingênua, indicaria o algoritmo usando iteração como mais eficiente, por, em mais momentos no gráfico, este algoritmo estar mais baixo que o gráfico do algoritmo usando recursão.

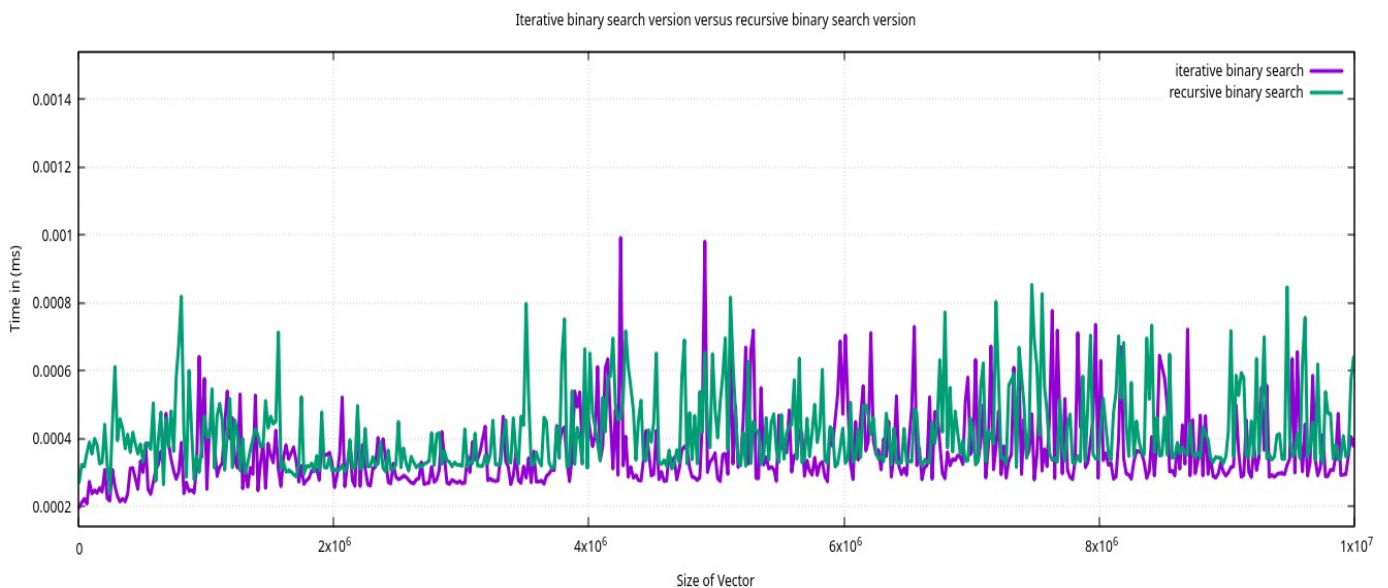


Figura 2: gráfico de comparação de desempenho da busca binária usando recursão e iteração