Análise de busca sequencial, sequencial otimizada e binária

Nínive Silva[1]

[1]Instituto Federal de Brasília (IFB)
Taguatinga – DF – Brasil

Abstract. *Querying data has always been a necessity, be it form data, database or any system that stores data, with this emerging some data search methods from which computing can apply these methods to search for data. Some of the methods used are sequential and binary search, each search has its own method and its performance depends on the amount and complexity of the data. This work proposes an analysis of the performance of these search methods by carrying out an empirical analysis.*

Resumo. *Consultar dados sempre foi uma necessidade seja dados de formulário, banco de dados ou qualquer sistema que armazene dados, com isso surgiu alguns métodos de busca de dados do qual a computação consegue fazer a aplicação desses métodos. Alguns dos métodos existentes é a busca sequencial e binária, cada busca tem seu próprio método e seu desempenho depende da quantidade e da complexidade dos dados. Este trabalho propõe uma análise do desempenho desses métodos de busca realizando uma análise empírica.*

1. Introdução

A computação lida com vários tipos de dados, esses dados precisam ser buscados/consultados seja em banco de dados ou em arquivos de formulário, sites e outros sistemas que armazenam dados, com isso surgiu alguns algoritmos de buscas. Alguns desses algoritmos de busca são; busca sequencial, busca binaria, ambos os algoritmos possibilita a busca por registros.

Esses algoritmos lidam com o método de buscar valores sobre uma pesquisa fornecida, verifica se o dado buscado consta nos registros e devolver o valor caso seja encontrado. Esse é um conceito geral sobre os métodos de busca, mas cada um tem segue uma particularidade no seu método para achar valores. O método de busca sequencial é o mais simples e menos otimizado, já o de busca binária é um método mais eficiente.

Neste artigo será apresentado uma análise de desempenho dos trés métodos de busca, busca sequencial, busca sequencial otimizada e a busca binária, cada método vai ter um código construído seguindo a estrutura do método e analisado individualmente.

Para os testes das analises vai ser utilizado valores representando registros e valores para serem buscados nos registros, medindo o tempo que cada método de busca leva para encontrar o valor procurado.

2. Fundamentação Teórica

Para essa pesquisa foram utilizados trés métodos, busca sequencial, busca sequencial otimizada e busca binária.

2.1. Método de Ordenação Sort

Alguns métodos de busca citados na introdução necessitam que os números procurados sejam ordenados para que o método funcione, para essa pesquisa é utilizado o método de ordenação próprio que vem embutido na linguagem escolhida para desenvolver os códigos.

O método de ordenação utilizado é o sort, este método de ordenação consiste em comparar o numero anterior com o próximo, caso o numero próximo numero seja maior ele é colocado na posição do anterior, caso não seja maior ele permanece na mesma posição.

2.2. Busca Sequencial

A busca sequencial é um método de busca simples, podendo ser aplicada onde exista uma sequencia de dados, esses dados podem ser de diferentes tipos como; letras, números ou uma lista com vários elementos que contem diversas informações.

Para que um elemento seja encontrado esse método trabalha verificando os dados armazenados um a um comparando com o registro procurado, começando do inicio dos dados até o final, caso seja encontrado volta para o inicio da lista de dados, até que todos os registros a serem procurados tenha sido verificado nos dados, após procurar todos os registros solicitados na busca, caso sejam encontrados nos dados o método devolve os dados encontrados.

Segundo [Alves], citado por [Ziviani et al. 2004] ao analisar a busca sequencial concluiu que o número esperado de comparações para verificar se um dado procurado está nos registro são: (elemento na primeira posição) o algoritmo possui complexidade O(1), no caso médio (elemento na posição central), O(n/2), e no pior caso (elemento na última posição), O(n)

2.3. Busca Sequencial Otimizada

A busca sequencial otimizada segue o mesmo método de busca da busca sequencial, mas com uma otimização, na busca sequencial otimizada caso valores não esteja em ordem crescente é feito uma ordenação desses dados e depois o processamento de busca do registro sobre esses dados, ter os dados ordenados em ordem crescente possibilita uma otimização no processo de busca, se o registro ao ser comparado com o valor procurado for maior, a busca volta para o inicio dos valores e busca um próximo dado a ser procurado. Essa otimização diminui o tempo de busca, pois se o valor procurar for menor que o do registro, então o dado procurado não está nos registros.

Porém, um ponto negativo dessa otimização é o tempo gasto para ordenar, pois além do tempo de busca tem o tempo de ordenação antes, caso seja muito registros essa ordenação pode levar algumas horas ou dias.

2.4. Busca Binária

O método de busca binária é diferente do método de busca sequencial, enquanto o método de busca sequencial busca um determinado registro sobre os dados de forma linear e seguindo uma sequencia, o busca binária compara o valor buscado com o dado central, ignorando a outra parte dos dados e diminuindo o tempo de busca.

A busca binária compara se o numero buscado é maior ou menor que o dado central, se caso o numero buscado for menor ele volta a busca pesquisando do inicio dos dados, se for maior ele parte do central até o final da lista, porém para realizar a busca binária assim como a busca otimizada, este método também precisa estar ordenado em ordem crescente. Segundo [Alves], citado por [Ziviani et al. 2004] a complexidade é sempre é O(log n)

Ele consiste em comparar inicialmente o elemento buscado com o elemento central do conjunto, de forma que uma das metades possa ser desprezada, minimizando o escopo de busca. Nessa comparação, verifica-se se o elemento buscado é maior ou menor que o elemento central, para que a busca parta para uma das metades. Para que isso aconteça, o conjunto de elementos a ser analisado deve estar em ordem crescente.

2.5. Análise Empírica

A análise empírica é um forma científica de coletar dados que podem ser mensuráveis, está análise se baseia em dados concretos em vez de teorias ou ideias. Essa análise aplica métodos e técnicas de pesquisa sobre os dados coletados, com o objetivo de buscar respostas reais.

3. Ferramentas e Tecnologias Utilizadas

Para o desenvolvimento dos testes foi utilizado algumas ferramentas e tecnologias para a criação dos códigos que realiza os métodos de busca.

3.0.1. Python

Para o desenvolvimento do código foi utilizado a linguagem Python, uma linguagem de alto nível; na programação alto nível é quando a linguagem se assemelha a escrita humana.

O Python é uma linguagem de sintaxe simples e fácil de ser compreendida, Python trabalho com vários tipos de paradigmas de programação; paradigmas é a estrutura que o código pode ser escrito. No Python os paradigmas mais utilizados são o funcional e o orientado a objetos. Para o desenvolvimento do código desse trabalho, foi utilizado o paradigma funcional.

3.0.2. Paradigma Funcional

O paradigma funcional citado na seção anterior foi utilizado para a construção do código dos métodos de buscas. Para [Jungthon and Goulart 2009] este paradigma trata a computação como uma avaliação de funções matemáticas. Este método enfatiza a aplicação de funções, as quais são tratadas como valores de primeira importância, ou seja, funções podem ser parâmetros ou valores de entrada para outras funções e podem ser os valores de retorno ou saída de uma função.

A utilização desse paradigma no código facilita o aproveitamento de uma bloco de código sem precisar cria-lo novamente, sempre que uma parte do código precisa de uma funcionalidade em que exista uma função criada, essa função pode ser chamada dentro do novo bloco de código.

3.0.3. Bibliotecas Utilizadas

Bibliotecas são um conjunto de módulos pre definidos e prontos que toda linguagem de programação possui. Para o desenvolvimento das análises foi utilizadas as da linguagem Python, pois foi a linguagem escolhida para o desenvolvimento do código dos métodos das análises.

- Numpy biblioteca utilizada para a manipulação e também operações de listas, vetores e matrizes.
- Pandas; biblioteca para a manipulação e análise de dados. oferece estruturas para manipulação de tabelas numéricas e manipulação de dados em vários tipos de formato.
- Seaborn biblioteca para criar gráficos, visualizações estatísticas e interativas.;
- Matplotlib biblioteca para criar gráficos, visualizações estatísticas e interativas.;
- Time; biblioteca que possibilita manipular, hora, data, e duração de tempo e intervalos de tempo.

4. Aplicação da Análise Empírica

A análise utilizada para os testes foi a empírica, seguindo a etapa de coleta e processando dos dados, a análise foi aplicada nos métodos de busca busca sequencial, sequencial otimizada e sequencial binária. Para cada teste foram utilizados quadro tipos de valores, todos os valores foram processados em uma função desenvolvida com os métodos de busca, para a realização das análises.

Esta análise segue etapas que auxiliam na aplicação que são;

- Definir o objetivo da investigação
- Apoiar-se em teorias relevantes
- Criação de hipóteses e medição
- Metodologia, desenho da pesquisa e coleta de dados
- Análise de dados e resultados da pesquisa empírica
- Conclusão

4.1. Dados Utilizados

Está etapa de coleta dos dados corresponderia a etapa de coleta de dados da análise empírica são valores gerados aleatórios para n e q, os valores de n são entre n= 10^4, n= 10^5, n= 10^6, n= 10^7 gerados aleatórios nesta quantidade e neste intervalo de números. Para q os valores são no intervalo de q= 10^2, n= 10^3, n= 10^4, n= 10^5 nesta quantidade e desse intervalo de números.

4.1.1. Aplicação dos Testes

No desenvolvimento do código para os métodos de busca sequencial, sequencial otimizada e binária, foi utilizado o paradigma funcional, criando uma função para cada busca, a função foi nomeada com os nomes das buscas, cada uma contendo sua funcionalidade, citado no capitulo acima em Fundamentação Teórica.

Após a criação das funções, foi criado uma função main para a chamada dessas funções, o arquivo main na linguagem Python é o principal, ele é o primeiro a ser processado no código, no código esse main ficou responsável pela chamada de cada função de busca.

Para a realização dos testes foram utilizados valores n= 10ˆ4, n= 10ˆ5, n= 10ˆ6, n= 10ˆ7 onde n é a sequencia de números para encontrar o numero requerido e q= 10ˆ2, q= 10ˆ3, q= 10ˆ4, q= 10ˆ5 a quantidade de números requeridos. Para cada q=10ˆ é buscando ocorrências de números idênticos sobre n=10ˆ.

Depois da finalização de cada busca é armazenado n onde n é numero de busca e q é os números a serem buscados, também é armazenado o tempo que cada função leva para achar os valores buscados para cada n e q. Com esses valores armazenados foi criado o gráfico para medir o desempenho de cada função, o valor de busca n sobre o numero requerido q, cada função tem seu gráfico sobre cada valor n e q.

5. Desempenho das Análises

Com a realização dos testes em cada função de busca, foi gerado um relatório sobre o desempenho. Cada função teve seus testes realizados de forma individual.

5.1. Teste Busca Sequencial

A (Figura 3) mostra o desempenho da busca sequencial sobre os valores de busca e o tempo que a função levou para encontrar os valores, para cada valor.

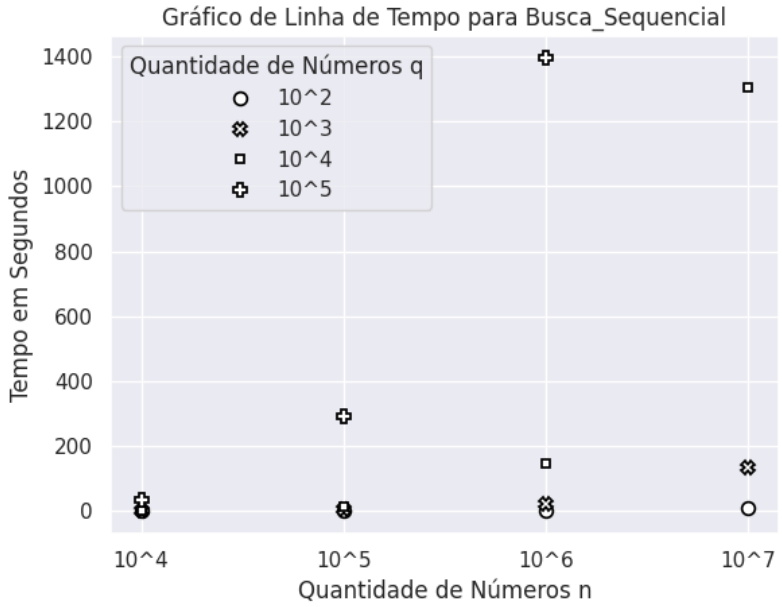

Figura 1. Gráfico da Análise de Busca Sequencial

No figura 1 que apresenta o gráfico é possível ver o comparativo tempo para cada valor n e q em segundos. Para valores em que n=10ˆ4 onde valores de q=10ˆ2 q = 10ˆ3, q = 10ˆ2, q = 10ˆ2 foram buscados sobre n mostra um desempenho rápido da função, para esse valores fica na faixa de 0 segundos e para e para valores em que n= 10ˆ4 a função

apenas tem um aumento quando o valor de q assume q= 10ˆ5, e para valores onde n= 10ˆ6 um aumento de tempo quando q também é q= 10ˆ5, porem um aumento um pouco mais significativo chegando a 1400 segundos. A complexidade aumenta quando os valores de n são n= 10ˆ7 onde tem uma variação maior do tempo.

A função busca sequencial se mostra bastante eficiente para valores em que n assume um intervalo de n= 10ˆ4 até n= 10ˆ7 quando q assume valores no intervalo de q= 10ˆ2 até q= 10ˆ4, mas quando n assume o valor n= 10ˆ10 e q= 10ˆ5 a função leva horas para finalizar. Então partir desses valores a função já diminui seu desempenho de forma significativa.

O Gráfico do desempenho da função de busca sequencial é representado pelo eixo x, que representa a quantidade de números da sequencia buscada que é o n e o q, eixo y representa a quantidade de tempo em segundos que a função levou para finalizar a busca.

5.2. Teste Busca Sequencial Otimizada

O método de busca sequencial foi testado para cada valor de n= 10ˆ, buscar dados dos valores q= 10ˆ sobre n.

A (Figura 3) mostra o desempenho da busca sequencial sobre os valores de busca e o tempo que a função levou para encontrar os valores, para cada valor.

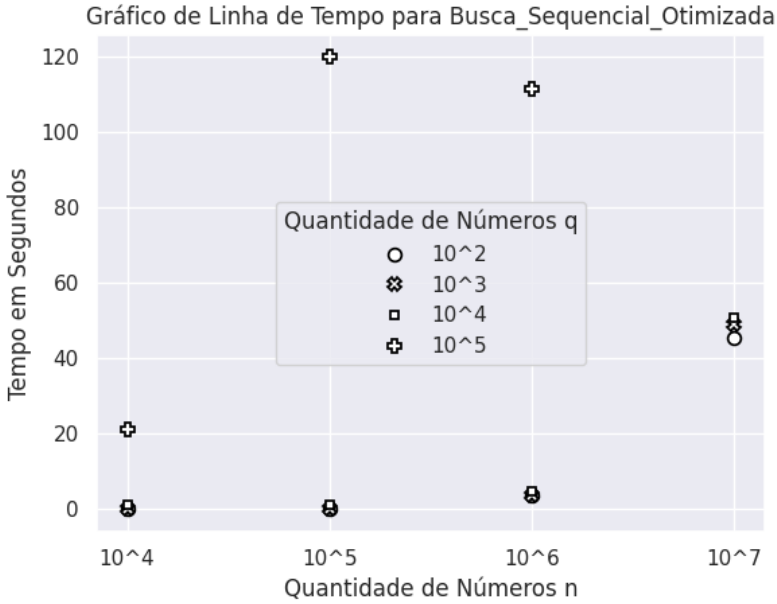

Figura 2. Gráfico da Análise de Busca Sequencial Otimizada

Na figura 2 que apresenta o gráfico é possível ver o comparativo tempo para cada valor n e q em segundos. A função busca sequencial otimizada apesar de ser otimizada em alguns casos a busca sequencial tem um desempenho melhor, mas isso é por causa do tempo de ordenação, o método de busca sequencial otimizado precisa que os valores de n que são os valores usados para serem pesquisados estejam ordenados em ordem crescente, então isso consome um gasto de tempo da função.

Para valores em que n=10ˆ4,onde valores de q = 10ˆ2, q = 10ˆ3, q = 10ˆ4, A função sequencial otimizada tem um desempenho igualável a da função sequencial, mas quando n = 10ˆ4 e q assume q = 10ˆ5 o desempenho da função sequencial otimizada leva 20 segundos de duração de busca contra na faixa de 0 da busca sequencial.

Para valores em que n= 10ˆ5 a função tem o desempenho melhor comparado função sequencial, quando q assume q= 10ˆ5. Então para valores grandes de q e n a função tem um bom desempenho, exceto quando n e q assumem valores onde n= 10ˆ7 e q= 10ˆ5 a função leva algumas horas para finalizar, a parti desses valores seria o ponto critico da função.

O Gráfico do desempenho da função de busca sequencial Otimizada é representado pelo eixo x, que representa a quantidade de números da sequencia buscada que é o n e o q, eixo y representa a quantidade de tempo em segundos que a função levou para finalizar a busca.

5.3. Teste Busca Binária

O método de busca binária foi testado para cada valor de n= 10ˆ, buscar dados dos valores q= 10ˆ sobre n.

A (Figura 3) mostra o desempenho da busca binária sobre os valores de busca e o tempo que a função levou para encontrar os valores, para cada valor.

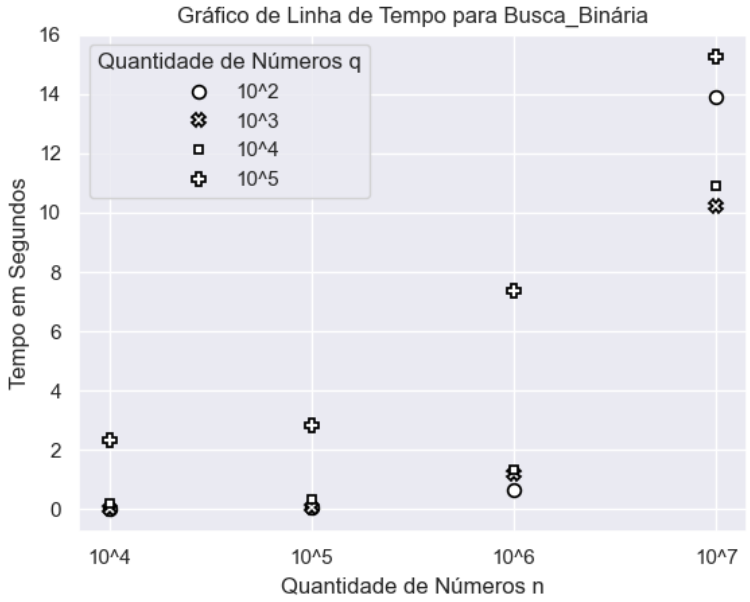

Figura 3. Gráfico da Análise de Busca Binária

A busca binária é a que tem o melhor desempenho independente dos valores assumidos por n e q, mesmo com a ordenação pois este método de busca binaria precisa que os valores de n sejam ordenados em ordem crescente, apresentou um excelente desempenho ficando na faixa de segundos de 0 até o máximo 0.10, mesmo quando n e q assumiam valores n= 10ˆ7 e q n= 10ˆ6 ou n= 10ˆ7 e q= 10ˆ7.

Para valores em que seriam o ponto critico das funções sequencial e sequencial otimizada, a busca binária teve um desempenho muito superior.

Para encontrar o ponto critico de desempenho da binária teria que realizar testes com n e q maiores dos que já foram testados.

O Gráfico do desempenho da função de busca sequencial é representado pelo eixo x, que representa a quantidade de números da sequencia buscada que é o n e o q, eixo y representa a quantidade de tempo em segundos que a função levou para finalizar a busca.

6. Configurações do Computador

Os testes foram realizados em um computador com dual Boot contendo sistema operacional Windows 10 e Linux Ubuntu, as configurações do computador são; processador Intel core i3 com 16GB de memoria RAM.

7. Desafios

Alguns dos desafios que surgiram no desenvolvimento dos métodos foi o tempo de processamento quando os valores eram números n= 10ˆ7 q= 10ˆ5 que são os números maiores em que n e q podem assumir, onde n é a quantidade de valores a serem buscados e q a quantidade de números procurados nos valores de n.

8. Conclusões

Neste tópico será apresentado conclusões a respeito da análise empírica dos métodos de busca.

Para valores onde o intervalo é n=10ˆ5 até e q assume q = 10ˆ5 a busca sequencial é melhor o desempenho comparada a otimizada e apresentando quase mesmo desempenho da binária, então para valores nesta faixa a sequencial é facilmente aplicável. Porem, para valores superior essa faixa a busca sequencial se torna melhor aplicável, exceto quando assume valores onde n=10ˆ7 e q= 10ˆ5 a função leva algumas horas para finalizar, a parti desses valores o ideal será utilizar a função binária, pois ela apresenta melhor desempenho para números grandes.

Com está análise podemos ver que dependendo do valor cada função tem um desempenho e se o valores não são grandes é possível aplicar o método sequencial e em certa faixa de números a sequencial otimizada é melhor para ser aplicada. Porem, para números grandes é binária é a indicada pelo seu desempenho em realizar em um tempo muito curto.

9. References

[Alves] [Jungthon and Goulart 2009] [Ziviani et al. 2004]

Referências

Alves, J. R. Análise de desempenho dos algoritmos de busca sequencial e de busca binária como ferramentas de um sistema de gerenciamento bancário.

Jungthon, G. and Goulart, C. M. (2009). Paradigmas de programação. *Monografia (Monografia)—Faculdade de Informática de Taquara, Rio Grande do Sul*, 57.

Ziviani, N. et al. (2004). *Projeto de algoritmos: com implementações em Pascal e C*, volume 2. Thomson Luton.