

UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE – UFRN
BACHARELADO EM TECNOLOGIA DA INFORMAÇÃO – BTI

Relatório: Análise empírica dos algoritmos de busca sequencial e binária,
implementados de forma recursiva e iterativa

Componentes:
Bianca Cristiane Ferreira Santiago
2016038862

Jaine Rannow Budke
2016035243

16 de março de 2017
Natal, RN

Relatório: análise empírica dos algoritmos de busca

A análise foi realizada tendo como base os algoritmos de busca sequencial e binária, implantados de forma recursiva e iterativa. O projeto foi implementado utilizando a linguagem C++ e o conceito de boas práticas de programação, como o uso do makefile para compilação do projeto modularizado e o uso do padrão Doxygen de documentação.

1. Organização do projeto

O projeto foi implementado com modularização em arquivos. Os arquivos .cpp estão localizados na pasta /src. O arquivo buscas.cpp é composto pela implementação dos quatro algoritmos de busca, enquanto o main.cpp possui funções para gerar bases de dados aleatórias e ordenadas, funções para fazer a medição do tempo dos algoritmos de busca - levando em consideração chaves de busca diferentes -, bem como a implementação de direcionar os dados para arquivos no formato txt.

2. Funcionamento do algoritmo

Ao executar o algoritmo, são gerados doze arquivos no formato txt. Grupos de tres arquivos são referentes a um mesmo tipo de busca:

- 1 - binária recursiva;
- 2 - binária iterativa;
- 3 - sequencial recursiva;
- 4 - sequencial iterativa.

Cada um desses tres arquivos possui relação com uma chave de busca diferente, que foram escolhidas de modo que:

1. Melhor caso sequencial: primeiro elemento do vetor da base de busca.
* base[0].
2. Pior caso sequencial e pior caso binária: valor superior ao último elemento do vetor de busca (elemento não pertencente ao vetor).
* base[tamBase-1]+1.
3. Melhor caso binária: elemento posicionado na metade do vetor.
* base[tamBase/2].

Dentro de cada um dos arquivos txt tem duas colunas com valores equivalentes ao tamanho da base de busca e o tempo de execução. Cada algoritmo de busca é executado com 14 tamanhos diferentes de base de busca = { 100, 500, 1000, 5000, 10000, 20000, 30000, 40000, 50000, 60000, 70000, 80000, 90000, 100000 } e o tempo de execução de cada algoritmo é medido por meio da função clock(), da biblioteca time.h.

A análise do algoritmo foi feita por meio de gráficos gerados através da ferramenta científica gnuplot. Os gráficos gerados possuem as coordenadas x e y representadas, respectivamente, como tamanho da base de busca e o tempo de execução dos algoritmos. Foram gerados quatro gráficos, cada qual equivalente a um tipo de busca.

3. Gráficos para análise empírica

A busca sequencial iterativa (Gráfico 1), independente do tamanho da base de busca, permaneceu com um tempo de execução muito próximo de 0 quando a chave de busca foi equivalente ao primeiro valor da base de dados, sendo, portanto, o melhor caso deste algoritmo. Quando a chave de busca foi alterada para o elemento do meio do vetor da base, o tempo de execução permaneceu na média para a faixa de valores de tamanho de busca analisada. Ao executar a busca procurando um valor não pertencente ao vetor, o algoritmo teve o pior desempenho, sendo caracterizado, portanto, como pior caso desta busca.

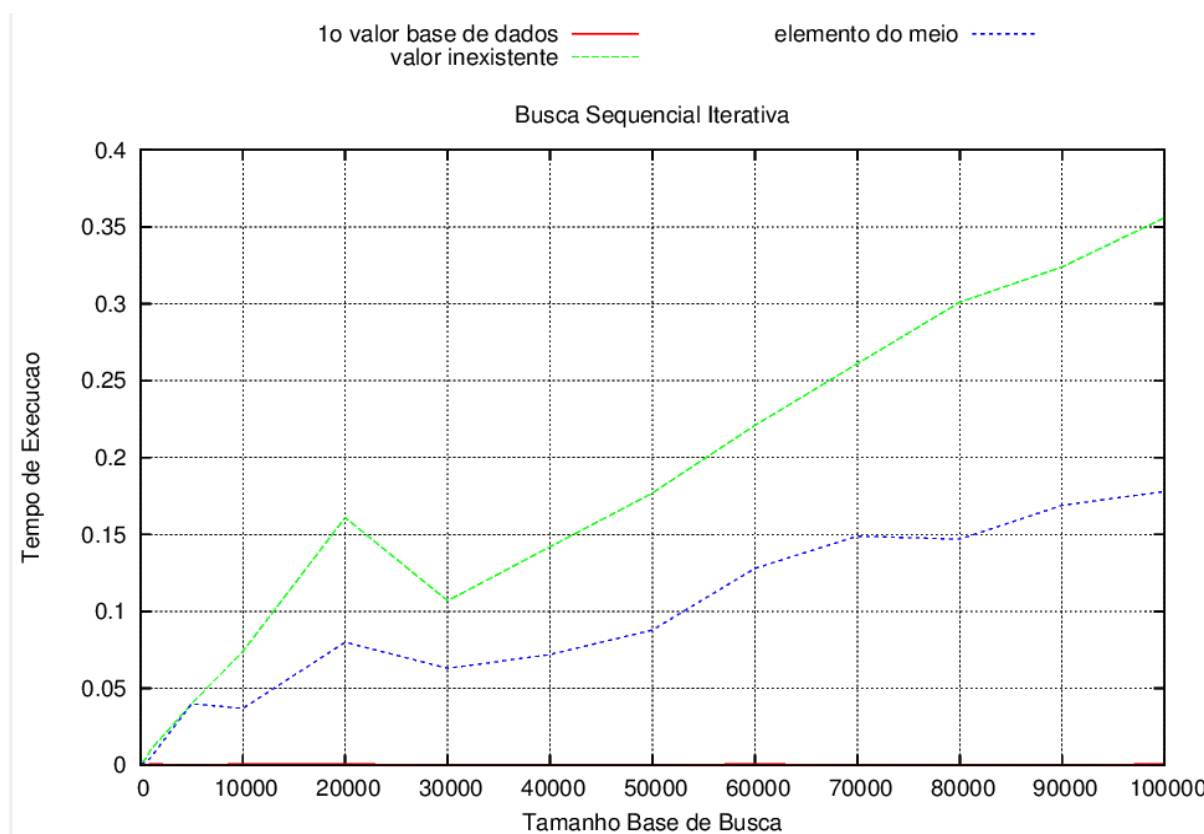


Gráfico 1 – Análise da busca sequencial iterativa

A busca sequencial recursiva (Gráfico 2) teve um desempenho semelhante à primeira. Para as chaves de busca analisadas, o pior, melhor e médio caso permaneceram os mesmos, alterando apenas os valores de tempo de execução.

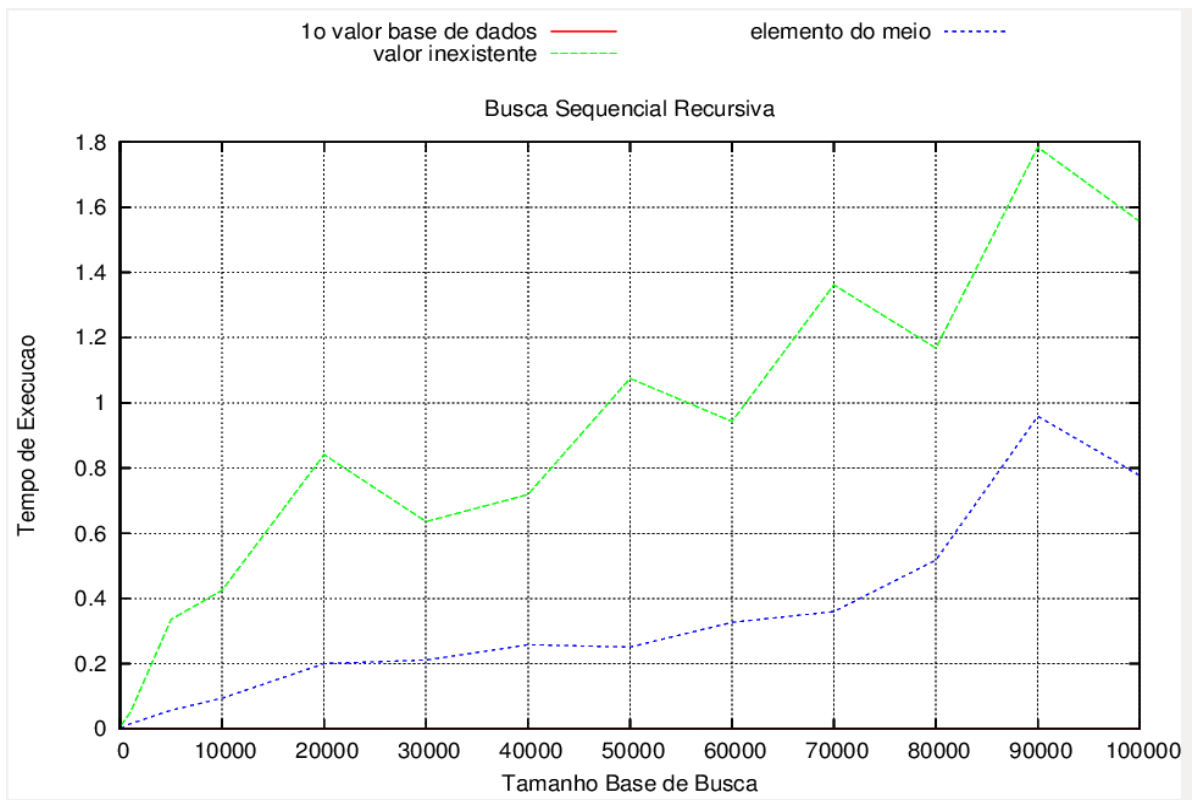


Gráfico 2 – Análise da busca sequencial recursiva

A busca binária recursiva (Gráfico 3) permaneceu com o menor tempo de execução quando a chave de busca foi equivalente ao elemento do meio da base de dados, sendo, portanto, o melhor caso deste algoritmo. Quando a chave de busca foi alterada para o primeiro elemento do vetor da base, o tempo de execução permaneceu na média para a faixa de valores de tamanho de busca analisada. Ao executar a busca procurando um valor não pertencente ao vetor, o algoritmo teve o pior desempenho, sendo caracterizado, portanto, como pior caso desta busca.

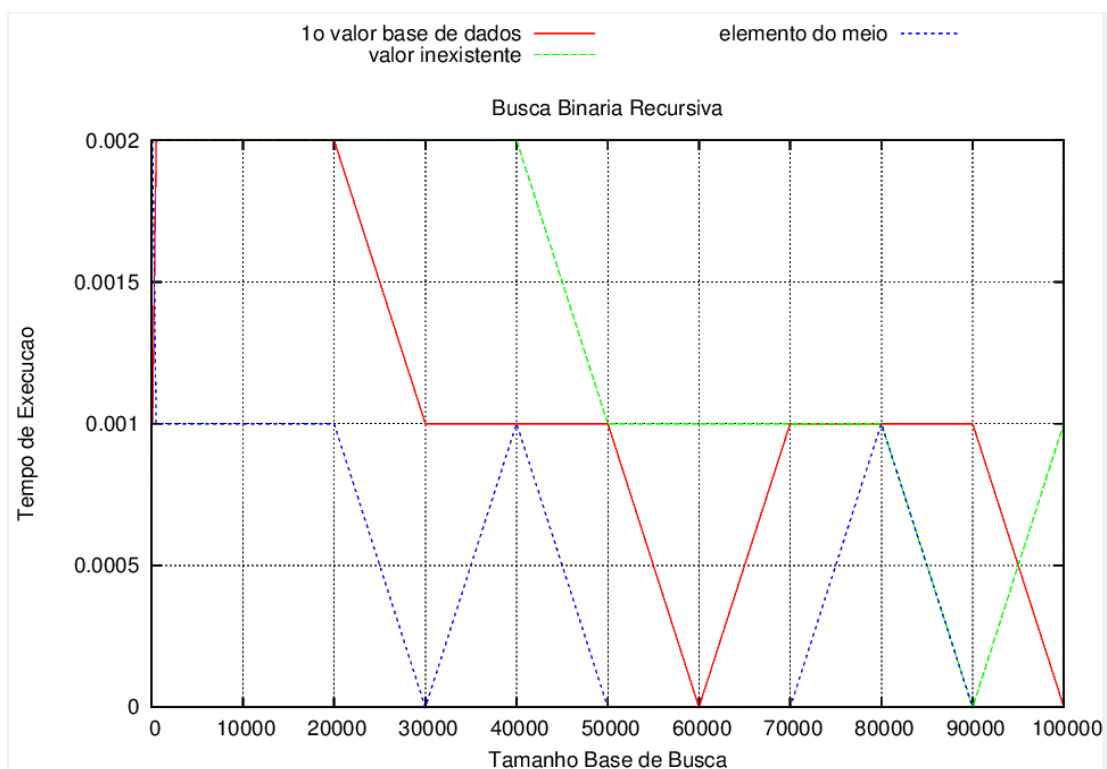


Gráfico 3 – Análise da busca binária recursiva

A busca binária iterativa (Gráfico 4) teve um desempenho semelhante à primeira. Para as chaves de busca analisadas, o pior, melhor e médio caso permaneceram os mesmos, alterando apenas os valores de tempo de execução.

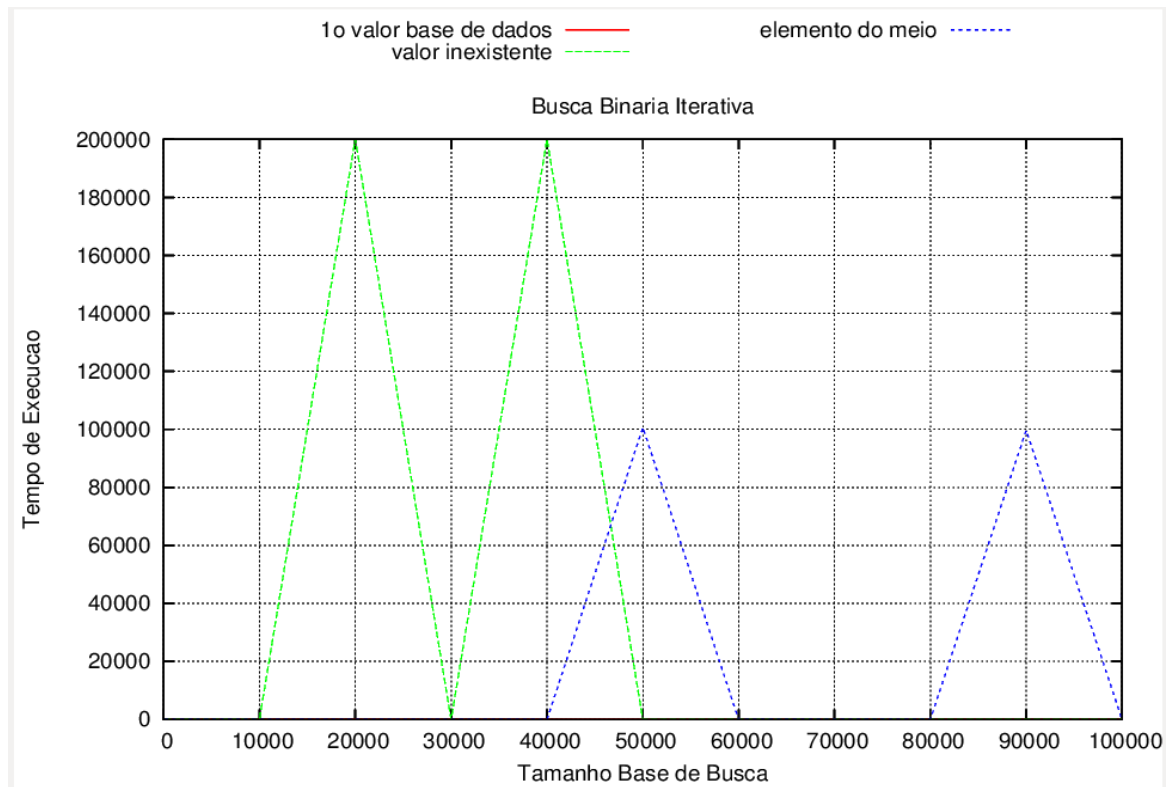


Gráfico 4 – Análise da busca binária iterativa