

Implementando Algoritmo de Busca e Análise Empírica

Natal-RN, 29 de junho de 2021

Prof. Selan Rodrigues

Aluno: Neylane Pereira Lopes

Matrícula: 20200038300

1 Introdução

O seguinte relatório tem como objetivo analisar os gráficos de tempo de execução que ilustram o comportamento dos algoritmos de busca, linear search, binary search e binary search recursive. Com intuito de responder experimentalmente perguntas como: "É realmente verdade que a busca binária é mais eficiente do que a busca linear?" e "Qual é mais eficiente, a versão iterativa ou recursiva da pesquisa binária?".

Para isso, foram implementados os algoritmos de busca e um sistema para capturar o tempo de execução deles, produzindo tabelas com amostras de dados para cada algoritmo. Depois de completar as tabelas, os dados foram usados para plotar os gráficos e, assim, auxiliar nas respostas das questões propostas.

2 Desenvolvimento

Inicialmente foram desenvolvidas e testadas, utilizando a linguagem de programação C++, as funções linear search, binary search, binary search recursive, lower bound e upper bound. Além disso, foi implementado um programa para capturar o tempo de execução dos algoritmos linear search, binary search e binary search recursive. Assim, foi executado cada algoritmo separadamente e capturado a taxa de crescimento de cada busca no pior caso, que corresponde a buscar por um elemento que não está no vetor, o elemento utilizado foi -1 .

A quantidade de testes realizados para cada algoritmo de busca foram no total 810, em que para cada tamanho de entrada eram realizados 10 testes, a fim de obter dados mais precisos. Dessa forma, os diferentes intervalos utilizados como entrada somam 81. Assim, para os três códigos de busca foram realizados 2.430 testes no total.

Desse modo, para cada algoritmo, foi aumentado progressivamente o tamanho do vetor de entrada adicionando $\frac{(10^8-10^4)}{80}$, isto é, 1.249.875 elementos de cada vez, iniciando com um vetor de 10^4 elementos até 10^8 elementos, e armazenando o tempo de execução dos 10 testes, de cada intervalo, em um arquivo formato csv. Em seguida, foi calculada a média de tempo de cada 10 testes, ou seja, gerando 81 médias para serem utilizadas na plotagem de um gráfico.

Para traçar os gráficos correspondente a cada algoritmo e calcular as médias foi utilizado como ferramenta o Google Colaboratory, além disso, a biblioteca matplotlib da linguagem Python e a biblioteca Pandas responsável pela manipulação dos dados.

2.1 Análise do tempo de execução de linear search versus binary search

Com base nas médias de tempo de execução geradas (Figura 1) foi possível plotar o gráfico da Figura 2, que contém a taxa de crescimento do linear search e do binary search.

	Tamanho de entrada (vetor)	Taxa de crescimento de Binary Search	Taxa de crescimento de Linear Search
0	10000	0.000145	0.052803
1	1259875	0.000204	2.997872
2	2509750	0.000217	4.490203
3	3759625	0.000225	6.394433
4	5009500	0.000213	8.691969
...
76	95000500	0.000282	168.839200
77	96250375	0.000272	171.057900
78	97500250	0.000282	172.629800
79	98750125	0.000278	174.920600
80	100000000	0.000275	177.155000

Figura 1: Tabela de linear search versus binary search

Observando o gráfico da Figura 2, percebe-se no último teste que o tempo de execução do linear search é superior a 175 milissegundos, enquanto o do binary search é aproximadamente 0 milissegundos. Analisando o gráfico, também é possível notar que quanto maior é a quantidade de elementos de entrada, maior é o tempo de execução do linear search em comparação ao do binary search.

Dessa maneira, no caso do algoritmo linear search a complexidade de tempo é $O(n)$, isto é, apresenta o comportamento de uma função linear, já o binary search a complexidade de tempo do algoritmo é $O(\log n)$, ou seja, tem o comportamento de uma função logarítima.

Portanto, o algoritmo binary search apresenta melhor desempenho que o linear search, assim, é realmente verdade que a busca binária é mais eficiente do que a busca linear, após analisar a taxa de crescimento de cada código.

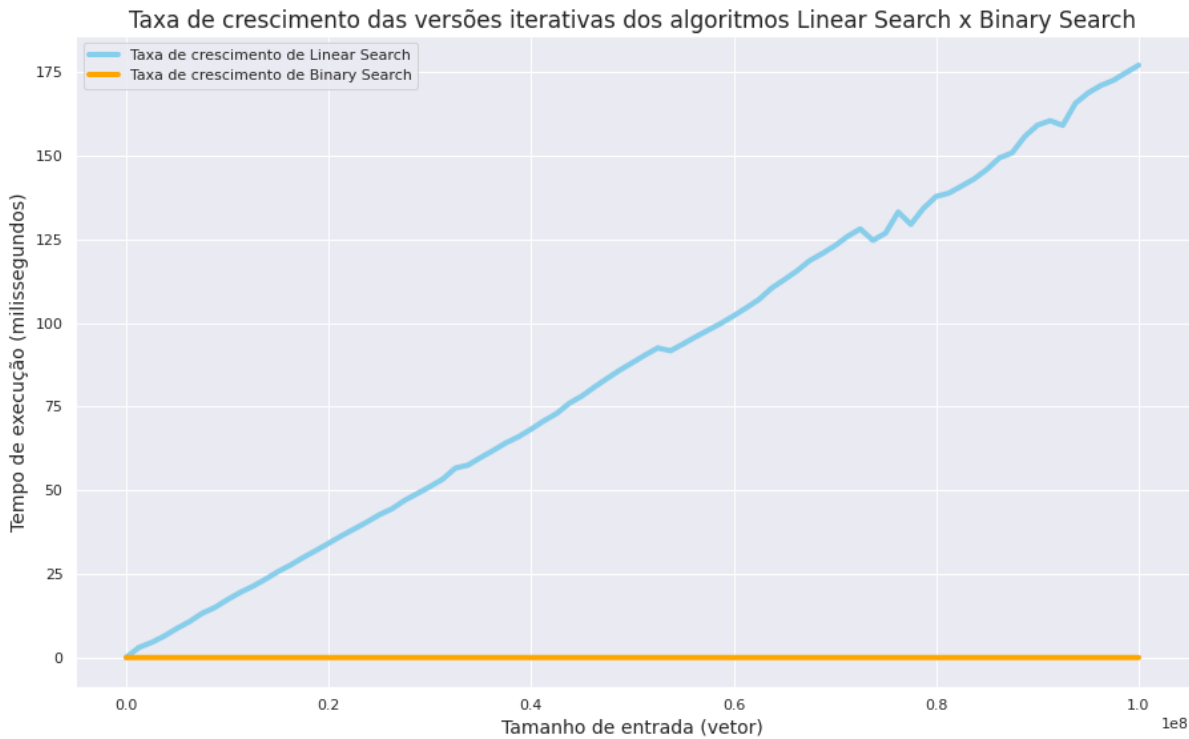


Figura 2: Linear search versus binary search

2.2 Análise do tempo de execução de binary search versus binary search recursive

Com base nas médias de tempo de execução geradas (Figura 3) foi possível plotar o gráfico da Figura 4, que contém a taxa de crescimento do binary search e do binary search recursive.

Tamanho de entrada (vetor)	Taxa de crescimento de Binary Search	Taxa de crescimento de Binary Search Recursive	
0	10000	0.000145	0.000193
1	1259875	0.000204	0.000267
2	2509750	0.000217	0.000276
3	3759625	0.000225	0.000279
4	5009500	0.000213	0.000290
...
76	95000500	0.000282	0.000417
77	96250375	0.000272	0.000494
78	97500250	0.000282	0.000358
79	98750125	0.000278	0.000426
80	100000000	0.000275	0.000396

Figura 3: Tabela de binary search versus binary search recursive

Analisando o gráfico da Figura 4, percebe-se no último teste que o tempo de execução do binary search está entre 0,00025 e 0,00030 milissegundos, enquanto o do binary search recursive é aproximadamente 0,00040 milissegundos. Observando os demais pontos do gráfico, também é possível notar que em todos os testes realizados o tempo de execução do binary search recursive é superior ao tempo do binary search.

Desse modo, tanto o algoritmo binary search quanto o binary search recursive possuem a complexidade de tempo $O(\log n)$, ou seja, tem o comportamento de uma função logarítmica. Porém, é perceptível que o algoritmo binary search apresenta melhor desempenho que o binary search recursive, portanto, a busca binária iterativa é mais eficiente do que a busca binária recursiva, após analisar a taxa de crescimento de cada código.

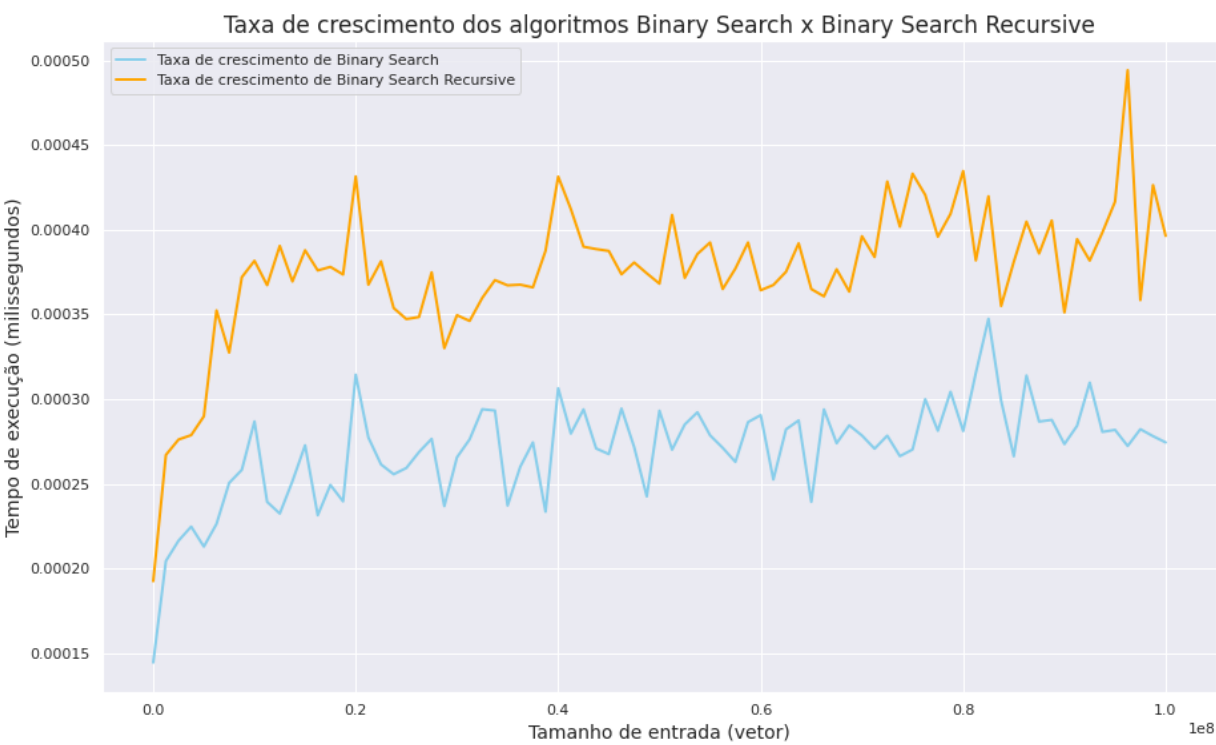


Figura 4: Binary search versus binary search recursive

3 Conclusão

Portanto, ao final dessas análises foi possível concluir que a busca binária iterativa é consideravelmente mais eficiente que as demais formas de busca, visto que ela tem um menor tempo de execução em relação a busca linear e a busca binária recursiva, sendo assim, uma das melhores formas de implementação para realização de buscas quando se trata de eficiência.