Relatório: LEDA - Análise de Algoritmos de Ordenação

Aluno: José Luiz Leite de Barros Neto – Turno: Noturno. 2024.1

Professor: Janderson Jason Barbosa Aguilar

1 INTRODUÇÃO

Este relatório é um requisito da disciplina de Laboratório de Estrutura de Dados (LEDA) para com a qual foi realizada uma análise comparativa descritiva de algoritmos de busca com base em suas relações de recorrência e características assintóticas.

Sendo assim, foi analisado os algoritmos de Busca Linear (B.L), Busca Linear de Duas Pontas (B.L.2), Busca Linear Recursiva (B.L.R), Busca Binária Iterativa (B.B.I) e Busca Binária Recursiva (B.B.R) avaliando sempre o pior caso – quando não há o elemento a ser buscado -, pois, o tempo de execução do pior caso de um algoritmo é o limite superior sobre o tempo para qualquer entrada.

O objetivo deste trabalho é comparar e descrever os algoritmos de busca e sua complexidade, além de determinar seus limites assintóticos superiores e buscar identificar quais deles seriam melhores em termos de tempo de execução.

2 METODOLOGIA

2.1 Extração de Dados

Durante o procedimento de extração de dados foi-se necessário criar uma Classe auxiliar para modificações que permitissem salvar os dados em um arquivo para, assim, analisá-los. Além disso, houve a criação de um laço para que a Classe “BrincandoComBusca” executasse em torno de 100 vezes. Dessa forma, é possível visualizar a distribuição do espaço amostral e comprovar que os dados estão corretamente distribuídos para a análise e não há alteração ou dados muito discrepantes.

2.2 Tecnologias para a Análise e Estruturação dos Dados

Após salvar os dados em um *arquivo.txt*, com a utilização da linguagem Python e a biblioteca Pandas foi criado um *DataFrame* – objeto bidimensional com formato de tabela - contendo, em cada coluna, os dados de tempo de execução dos cinco algoritmos de busca e, nas linhas, 100 repetições destes tempos.

2.3 Critérios de Análise

Devido a necessidade de planejar como seria feita a análise, foi realizada a definição de dois critérios essenciais para calcular as médias dos tempos de execução de cada algoritmo.

- A entrada de tamanho n – vetor ordenado – deve variar quanto a ordem de crescimento;

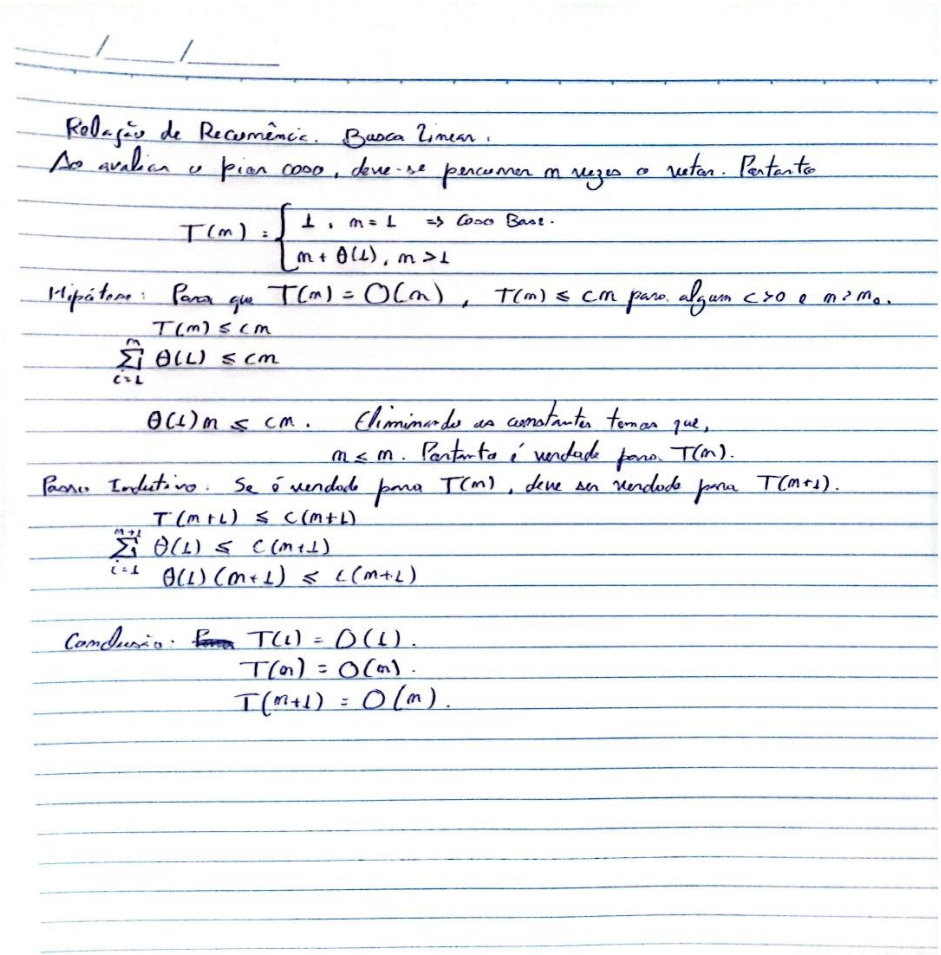
- Avaliar se os dados no DataFrame estão bem distribuídos.

Ambos os critérios se justificam pelo fato de que, primeiro, se a entrada não varia não há como visualizar como o algoritmo se comportaria e, em segundo, se os dados fossem mal distribuídos significaria que há um alto desvio padrão, portanto os valores se distanciariam rapidamente da mediana e média inviabilizando a compreensão da eficiência algorítmica.

3 RESULTADOS

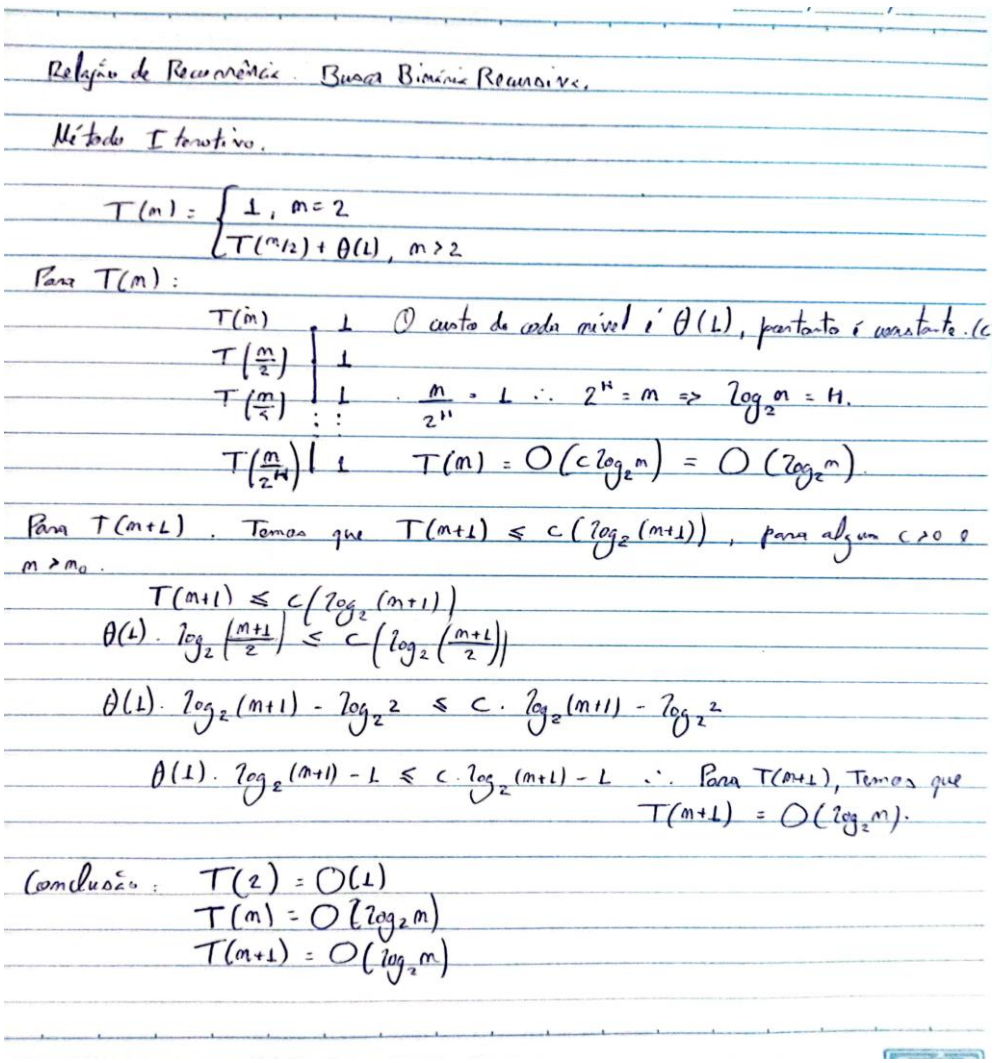
3.1 Relações de Recorrência e Notação Assintótica

Figura 1. Relação de Recorrência da Busca Linear determinando sua complexidade O(n).



Como a B.L, B.L.2 e B.L.R se trata de percorrer o vetor ordenado no pior caso, o limite superior para uma entrada n será O(n).

Figura 2. Relação de Recorrência da Busca Binária Recursiva com complexidade O (log n).



Os algoritmos B.B.R e B.B.I possuem a mesma característica assintótica, O (log n). A diferença é o modo iterativo e recursivo; e seus efeitos podem ser descritos na Tabela 1, assim como os outros algoritmos.

3.2 Cálculo das Médias dos Tempos de Execução

Tabela 1. A média do Tempo de Execução (em nanosegundos) dos cinco algoritmos de busca variando o n em 100, 1000, 10000 e 14000.

Tamanho	B. L	B.L.2	B.L. R	B.B. I	B.B. R
100	3058.0	1969.0	2906.0	690.0	1027.0
1000	20556.0	18525.0	10907.0	1038.0	1036.0
10000	71151.0	54352.0	58537.0	1478.0	1221.0
14000	65781.0	54756.0	55505.0	1408.0	973.0

É possível observar que com o aumento da entrada, B.L se comportou de maneira inconsistente – o tempo de execução cresce de maneira linear - comparado a B.B.R ou B.B.I que se manteve rápido e estável. Tal comportamento, tanto linear quanto logaritmo, é devido a complexidade do algoritmo. Pode ser comprovado que B.B.I e B.B.R são sempre superiores, devido sua característica assintótica. Nesse sentido, é determinado um limite superior log n para qualquer entrada n para o algoritmo de Busca Binária (Iterativo ou Recursivo).

Vale salientar que, há algumas diferenças em relação ao modo linear (B.L, B.L.2 e B.L.R) e logaritmo (B.B.I e B.B.R) quanto ao design dos algoritmos. A B.L.2, por exemplo, percorre o vetor ao mesmo tempo pelo início e pelo fim checando cada posição, dessa maneira se torna um pouco mais rápido que a B.L. Entretanto, B.L.2 e B.L.R são bem constantes em relação de um ao outro. Da mesma forma, os algoritmos B.B.I e B.B.R não são tão distantes entre si, pois ambos trabalham em tempo log n. No tamanho 100, B.B.I foi superior ao modelo recursivo, porém, quando n cresceu exponencialmente, B.B.R foi o melhor.

Durante os experimentos variando o tamanho da entrada, foi determinado o valor máximo de 14000, pois acima disso ocorria um “estouro da pilha” quando era realizado os algoritmos recursivos. Tal fenômeno é explicado por causa das recursões que são empilhadas na memória e excedem o limite permitido, ou seja, os algoritmos de B.B.R e B.L.R devem corresponder a um limite permitido pela pilha de recursões.

3. 3 Análise do Melhor Caso e Caso Médio

A experimentação do Melhor Caso (elemento na primeira posição para algoritmos lineares) revelou que para a B.L, B.L.2 e B.L.R tiveram melhor desempenho que a B.B.I e B.B.R. Pois, a busca binária realiza divisões para encontrar o elemento, já a busca linear simplesmente percorre o vetor. Utilizando um vetor de tamanho 14000, foi necessário, em média, de 500 nanosegundos para a B.L achar o elemento, mas a B.B.R, por exemplo, a média foi de 1122 nanosegundos. O Melhor Caso é subjetivo ao tipo de algoritmo utilizado, portanto para a B.B.I e B.B.R o Melhor Caso é o elemento está na metade do vetor. A experimentação obtida por avaliar o Melhor Caso da busca binária foi que, em média, B.B.R necessitou de 470 nanosegundos e a B.L teve a média de 42473 nanosegundos em um vetor de 14000 elementos.

Devido as subjetividades do design dos algoritmos, a avaliação do Caso Médio é onde podemos inferir qual algoritmo seria mais eficiente que o outro. Ainda utilizando um vetor de tamanho 14000, foi determinado elementos aleatórios para serem buscados e a B.L, por exemplo, teve uma média de 27440 nanosegundos para encontrar, já a B.B.R teve em média 1809 nanosegundos. Isto revela a consistência e eficiência da B.B.R ao buscar qualquer elemento em um vetor de tamanho n em comparação com a B.L.

4 CONCLUSÃO

Ao avaliar as relações de recorrência, características assintóticas de algoritmos e seu design, é possível implementar o melhor custo para uma entrada n, ou seja, o melhor tempo. Nesse sentido, saber determinar o limite superior de um algoritmo, assim como sua relação de recorrência, é essencial para compreender a eficiência algorítmica. Porém, como foi visto nos

resultados, o design e modificação de algoritmos já estabelecidos podem gerar uma eficiência melhor ou pior, dependendo do tipo de modificação. Portanto, é partir da análise assintótica de algoritmos que podemos implementar variações de código para buscar uma eficiência almejada para o tipo de problema a ser resolvido.