



UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE
INSTITUTO METRÓPOLE DIGITAL
BACHARELADO EM TECNOLOGIA DA INFORMAÇÃO

KARINA MARIA BEZERRA JACINTO

ANÁLISE EMPÍRICA DOS ALGORITMOS DE BUSCA LINEAR E BINÁRIA

NATAL, RN

2020

ANÁLISE EMPÍRICA DOS ALGORITMOS DE BUSCA LINEAR E BINÁRIA

KARINA MARIA BEZERRA JACINTO

Relatório técnico apresentado à disciplina Estrutura de Dados Básicas I, como requisito parcial para obtenção de nota da primeira unidade.

Docente: Selan Rodrigues dos Santos

Natal, RN

2020

LISTA DE ILUSTRAÇÕES

QUADRO

Quadro 1: Configurações da máquina utilizada	5
---	---

ALGORITMOS

Algoritmo 1: Configuração do vetor de buscas	6
Algoritmo 2: Busca linear iterativa	6
Algoritmo 3: Busca binária iterativa	7
Algoritmo 4: Busca linear recursiva	7

TABELA

Tabela 1: Relação entre tamanho da amostra e tempo de execução.....	8
--	---

GRÁFICOS

Gráfico 1: Comparação entre busca linear e busca binária	10
Gráfico 2: Comparação entre busca binária iterativa e recursiva.....	10

SUMÁRIO

1. INTRODUÇÃO	5
2. METODOLOGIA	5
3. RESULTADOS	8
4. DISCUSSÃO DOS RESULTADOS	11
5. CONCLUSÃO	11

1. INTRODUÇÃO

O uso de algoritmos de busca é feito de maneira recorrente na ciência da computação, pois quase sempre é necessário realizar buscas como pré-requisito para dar prosseguimento a determinadas tarefas. Logo, conhecê-los é importante para que possamos escolher um algoritmo que seja eficiente para cada situação.

Assim, o objetivo desse relatório é realizar uma análise empírica entre dois dos algoritmos de busca existentes, são eles o de busca linear e binária. Para a análise, o objetivo é responder as seguintes perguntas: a busca binária é mais eficiente que a busca linear? Qual a versão (recursiva ou iterativa) da busca binária é mais eficiente?

Esse relatório será seguido pela metodologia, onde serão descritos os materiais e métodos da pesquisa que nos levarão até as respostas. Logo após, serão exibidos os resultados para as perguntas que originaram o relatório. Em seguida, os resultados obtidos serão discutidos. E, por fim, será a conclusão.

2. METODOLOGIA

Todos os algoritmos descritos neste relatório foram implementados na linguagem C++ e compilados no g++ com o auxílio do cmake (versão 3.16.3). A máquina utilizada tem as configurações descritas na Tabela 1:

Quadro 1: Configurações da máquina utilizada

MÁQUINA	NOTEBOOK POSITIVO – 4GB RAM
PROCESSADOR	AMD® C-60 apu with radeon(tm) hd graphics × 2
ARQUITETURA	64 bits
SISTEMA OPERACIONAL	Ubuntu 20.04.1 LTS
GRÁFICOS	AMD® Palm

Fonte: Autoria própria

A contagem do tempo de duração de cada busca foi feita utilizando a biblioteca chrono. Para geração dos gráficos foi utilizado o gnuplot (versão 5.2). A configuração do vetor no qual as buscas serão realizadas pode ser visto no Algoritmo 1

Algoritmo 1: Configuração do vetor de buscas

```
int *array = new int[1000000000]; //109

for(int i{0}; i<1000000000; ++i){
    array[i] = i;
}
```

Fonte: Autoria própria

Apesar do vetor ser alocado com 10^9 elementos, a busca (linear ou binária) é feita inicialmente com os 10^8 primeiros elementos e a cada iteração o intervalo de busca é acrescido de dezoito milhões de números, o que nos permite realizar cinquenta buscas nesse vetor. É válido ressaltar que todas as análises dos algoritmos buscam reproduzir o pior caso, que é o fato do elemento procurado não existir no vetor. Diante disso, sempre buscaremos por valores negativos.

Prosseguindo, o primeiro algoritmo implementado foi o de busca linear iterativa, conforme podemos ver no Algoritmo 2. É importante dizer que os algoritmos mencionados a partir de agora usam o `value_type`, que é um tipo inteiro que foi denominado dessa forma no arquivo de cabeçalho das funções de busca.

Algoritmo 2: Busca linear iterativa

```
value_type * lsearch( value_type * first, value_type * last, value_type value )
{
    while(first != last && *first != value){
        first++;
    }
    return first;
}
```

Fonte: Autoria própria

Posteriormente, foi desenvolvida a busca binária iterativa (Algoritmo 3). Por último, foi elaborada a busca binária recursiva (Algoritmo 4)

Algoritmo 3: Busca binária iterativa

```

value_type * bsearch( value_type * first, value_type * last, value_type value )
{
    value_type step, *mid;
    value_type count = last-first;
    while(count != 0){
        step = (count)/2;
        mid = first+ step;
        if(*mid == value){
            return mid;
        }else if(*mid < value) {
            first = mid+1;
            count = count - (step+1);
        }
        else{
            count = step;
        }
    }
    return last;
}

```

Fonte: Autoria própria

Algoritmo 4: Busca linear recursiva

```

value_type * bsearchrecursive( value_type * first, value_type * last, value_type value )
{
    value_type size = last-first;
    value_type *middle = first+size/2;

    if(size != 0){
        if(*middle == value){
            return middle;
        }
        if(*middle > value){
            value_type *result = bsearchrecursive(first, middle, value);
            if(value == *result){
                return result;
            }
            return last;
        }

        return bsearchrecursive(middle+1, last, value);
    }
    return last;
}

```

Fonte: Autoria própria

3. RESULTADOS

A partir de agora serão demonstrados os resultados da análise empírica dos algoritmos de busca linear e binária feitos com uma amostra de 50 entradas variando entre 10^8 e 10^9 elementos. Observe que a Tabela 2 faz uma relação entre o tamanho do vetor de busca e o tempo que leva até que o algoritmo seja concluído no pior caso.

Tabela 1: Relação entre tamanho da amostra e tempo de execução

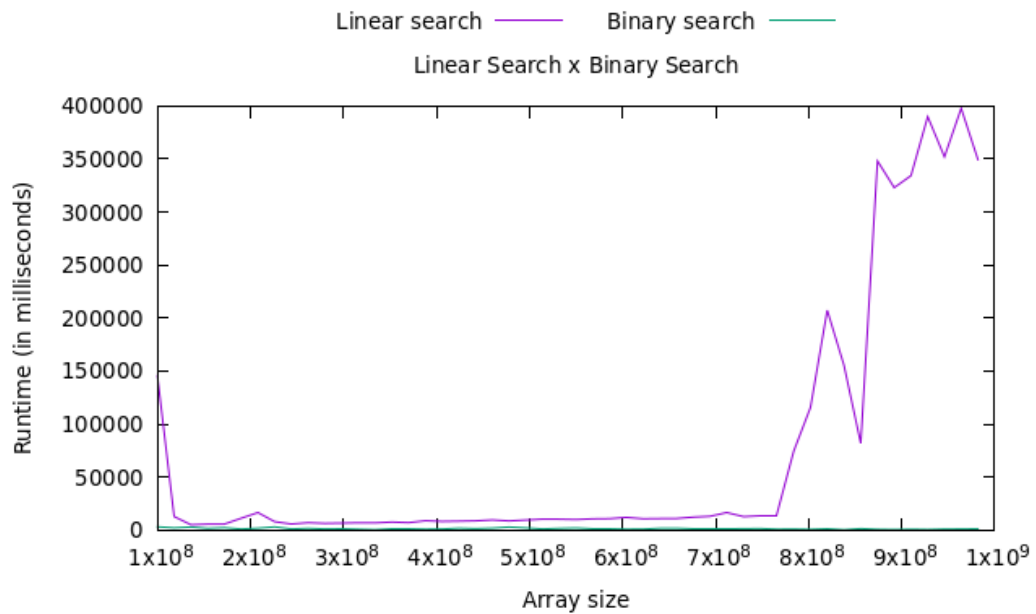
AMOSTRA	TIPO DE PESQUISA		
	LINEAR (ms)	BINÁRIA ITERATIVA (ms)	BINÁRIA RECURSIVA (ms)
100000000	145640	3157.53	103.68
118000000	12799.3	2120.21	663.736
136000000	5024.99	2928.78	1496.28
154000000	5598.34	1630.19	1642.73
172000000	5693.57	2207.99	1730.5
190000000	11192.4	1004.07	787.97
208000000	16536.5	1742.77	1314.96
226000000	7989.07	3065.52	1235.79
244000000	5756.27	1064.81	1876.92
262000000	6986.56	1641.01	1197.52
280000000	6384.79	1109.85	1365.09
298000000	6554.79	1192.85	1106.34
316000000	6872.75	748.299	2175.35
334000000	6806.86	388.659	2708.41
352000000	7604.64	802.038	2740.33
370000000	7061.4	761.401	1329.17
388000000	8836.79	843.222	753.976
406000000	8164.52	1009.31	754.403
424000000	8438.62	1729.12	2976.37
442000000	8721.9	1400.55	1371.9
460000000	9587.91	1826.83	1729.19
478000000	8633.97	2923.88	1581.41
496000000	9520.64	2062.74	1071.18
514000000	10382.5	1038.98	1517.85
532000000	10234.8	1600.95	1155.43
550000000	9950.12	1977.93	1630.57
568000000	10631.2	1027.9	2030.02
586000000	10819.2	1027.71	1752.51
604000000	11978.9	857.715	3283.8
622000000	10722.3	953.942	2018.9
640000000	10919	1681.88	1295.54
658000000	10981.6	1801.06	1100.35

676000000	12285.4	1260.29	1595.08
694000000	13066.9	759.376	1084.63
712000000	16553.3	748.929	2548.72
730000000	12749.9	1286.54	1724.28
748000000	12906.6	1492.2	1222.26
766000000	12900.6	819.19	1142.77
784000000	74458	920.503	720.913
802000000	115793	689.535	1038.53
820000000	207433	1251.46	1334.99
838000000	155503	334.819	2148.44
856000000	81866	1414.89	1165.76
874000000	348154	712.602	1123.51
892000000	323068	490.531	1940.44
910000000	334372	751.651	2042.73
928000000	390054	518.194	2013.95
946000000	352131	909.373	471.796
964000000	397890	1031.61	1580.25
982000000	349385	1156.65	792.486

Fonte: Autoria própria

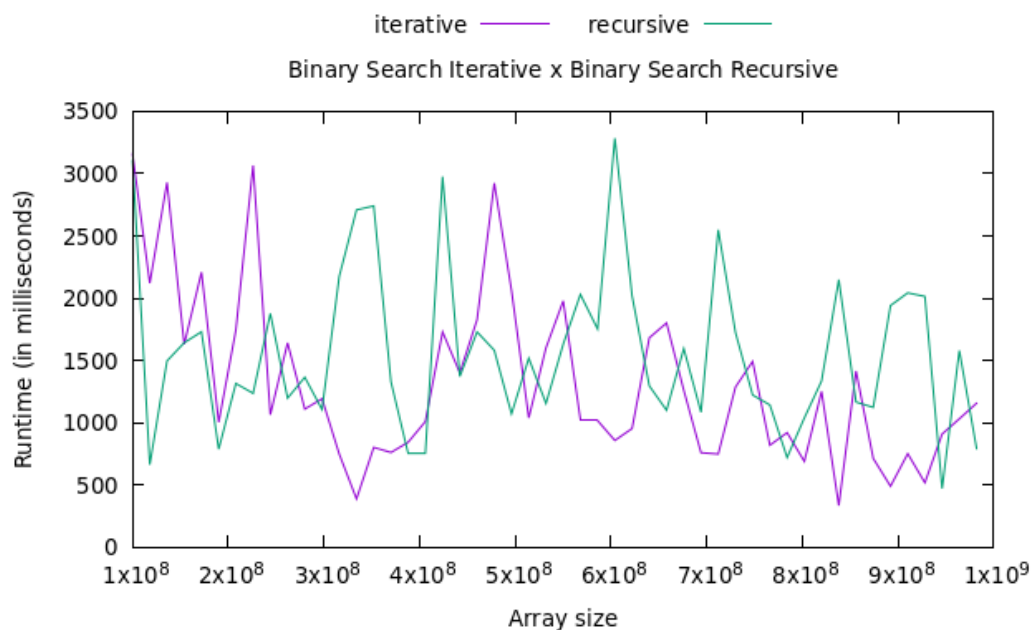
A partir dos dados tabelados, podemos gerar gráficos para visualizar o comportamento dos algoritmos e assim responder mais facilmente as perguntas: a busca binária é mais eficiente que a busca linear? Qual a versão (recursiva ou iterativa) da busca binária é mais eficiente?.

No Gráfico 1, podemos ver uma comparação entre a busca linear e a busca binária iterativa. Ao analisar o gráfico podemos perceber que a busca linear teve um crescimento linear nas primeiras buscas, mas depois do aumento gradativo da capacidade do array, o tempo de busca aumentou muito. Isso acontece porque a busca linear é um algoritmo de complexidade $O(n)$, isto é, o algoritmo realiza um número fixo de operações sobre cada elemento de entrada e como estamos analisando o pior caso, ocorre a verificação de todos os elementos do array para retornar que o elemento de busca não existe. Ainda no Gráfico 1, podemos ver que a busca binária se torna quase imperceptível em comparação com a busca linear, pois a busca binária segue métodos diferentes para encontrar o valor procurado. Ao contrário da busca linear, a binária realizar divisões no vetor de acordo com o valor que desejamos encontrar, logo a busca não é feita analisando elemento por elemento, o que permite que o tempo de busca seja bastante otimizado.

Gráfico 1: Comparação entre busca linear e busca binária

Fonte: Autoria própria

No Gráfico 2, é possível visualizar a comparação entre a busca binária iterativa e recursiva. Nele, fica visível que a partir de 6×10^8 elementos a busca iterativa é mais eficiente que a recursiva. Isso acontece devido ao fato de que a busca binária recursiva consome mais memória nas chamadas a diversos níveis de recursão, pois cada chamada aloca memória para os parâmetros e variáveis locais. Logo, quanto maior o vetor, mais chamadas recursivas são feitas, o que aumenta o tempo de execução do algoritmo.

Gráfico 2: Comparação entre busca binária iterativa e recursiva

Fonte: Autoria própria

Portanto, temos o resultado que a busca binária é mais eficiente que a busca linear. E também que a busca binária iterativa é mais eficiente que a busca binária na maior parte dos casos.

4. DISCUSSÃO DOS RESULTADOS

Ao analisar os dados da pesquisa chegamos à conclusão de que a busca binária iterativa é mais eficiente que a busca linear, considerando que o vetor de busca está ordenado. Como já comentando isso se deve ao fato da complexidade dos dois algoritmos para o pior caso.

Além disso, também podemos concluir que a busca binária iterativa é mais eficiente que a recursiva para vetores com um grande número de elementos, o que acontece devido ao alto consumo de memória

5. CONCLUSÃO

Nesse relatório foi feita a análise empírica dos algoritmos de busca linear e busca binária na sua versão iterativa e recursiva. Ao realizar a análise podemos, portanto, concluir que a busca binária é mais eficiente que a busca linear, considerando vetores ordenados. E, por fim, concluímos que a busca binária iterativa é mais eficiente que a busca binária recursiva.