

Universidade Federal do Rio Grande do Norte
Instituto Metr pole Digital
DIM0119 - ESTRUTURA DE DADOS B SICAS I
TURMA T01

Main Mission 01

Natal-RN, 20 de setembro de 2021

Prof. Selan Rodrigues dos Santos

Aluno: Jo o Guilherme Lopes Alves da Costa

Matr cula: 20200045609

Implementa  o de Algoritmos de Busca e An lise Emp rica

Depois de implementar os seguintes algoritmos de busca: *linear search*, *binary search*, *lower bound* e *upper bound*, foi requisitado que f z ssemos an lise emp rica entre alguns algoritmos. Essa an lise pode ser feita com o intuito de descobrir qual dos algoritmos finaliza suas opera  es mais r pido, ou seja, qual deles demora menos tempo de processamento.

Para fazer essa an lise foi criado um arquivo de teste .cpp em que chamamos a fun  o de cada algoritmo em seu pior caso (o caso em que n o encontramos o valor no vetor) 30 vezes e tiramos a m dia, come ando com um vetor (foi usado vector) com 10^4 elementos. Depois, adicionamos mais 20.000 elementos ao vetor e repetimos o processo para cada algoritmo, e fomos fazendo isso at  chegar em 10^8 elementos. E para marcar o tempo foi utilizado a biblioteca Chronos do C++. Al m disso, foi necess rio a manipula  o de streams para mandar os tempos de cada algoritmo para um arquivo de texto. Diante disso, busquei a resposta dos seguintes questionamentos:

1. "  realmente verdade que a busca bin ria   mais eficiente do que a busca linear?"

Depois de marcado o tempo de cada algoritmo, coletamos os dados da busca bin ria e comparamos com a da busca linear atrav s de um gr fico pelo gnuplot. O que p de ser notado   que o tempo da busca linear foi aumentando muito mais r pido do que a da busca bin ria. Pelo gr fico, a fun  o da busca bin ria parece at  que   constante, devido a escala adotada (a fun  o aparenta estar em cima do eixo x), por m, o que acontece   que ela tem uma taxa de crescimento bastante baixa ao contr rio da busca linear. No gr fico da busca linear podemos ver que seu gr fico   semelhante a uma fun  o linear ($f(x) = ax$), s  n o   mais pr xima ainda a uma fun  o linear devido  s oscila  es de tempo causada pelo computador. Devido a isso, podemos afirmar que busca linear possui um grau de complexidade $O(n)$, enquanto a busca bin ria possui um grau de complexidade $O(\log n)$, devido ao seu crescimento ser muito devagar. Logo, podemos afirmar que a busca bin ria   mais eficiente que a busca linear.

2. "Qual   mais eficiente, a vers o iterativa ou a recursiva da busca bin ria?"

Para a gera  o do gr fico desse t pico foi realizado o mesmo processo do t pico anterior. Apesar da curva da vers o iterativa da busca bin ria ser parecida com a sua vers o recursiva no gr fico, elas possuem uma diferen a not vel. Podemos perceber que a vers o iterativa   mais r pida do que a vers o recursiva, mesmo essa diferen a n o sendo gritante. Isso se deve porque na fun  o recursiva temos o empilhamento, e ao dar errado um teste devemos desempilh -la para fazer um novo teste, e isso acaba gastando maior tempo, enquanto na iterativa ela j  vai pro pr ximo la o. Desse modo, percebemos que a vers o iterativa da busca bin ria   mais eficiente do que a vers o recursiva. Apesar disso, ambas possuem complexidade $O(\log n)$.

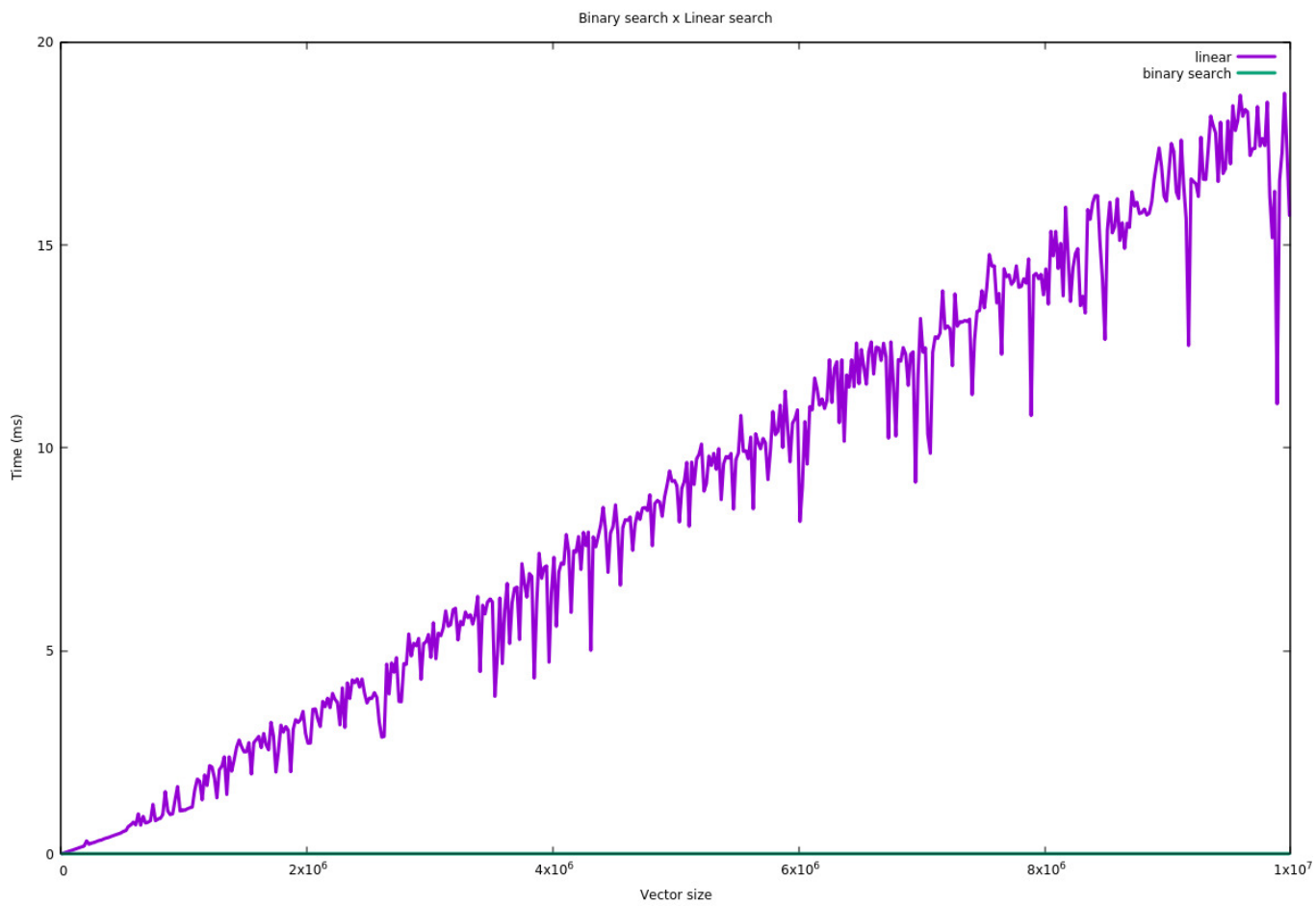


Figura 1: Gr fico referente ao item 1.

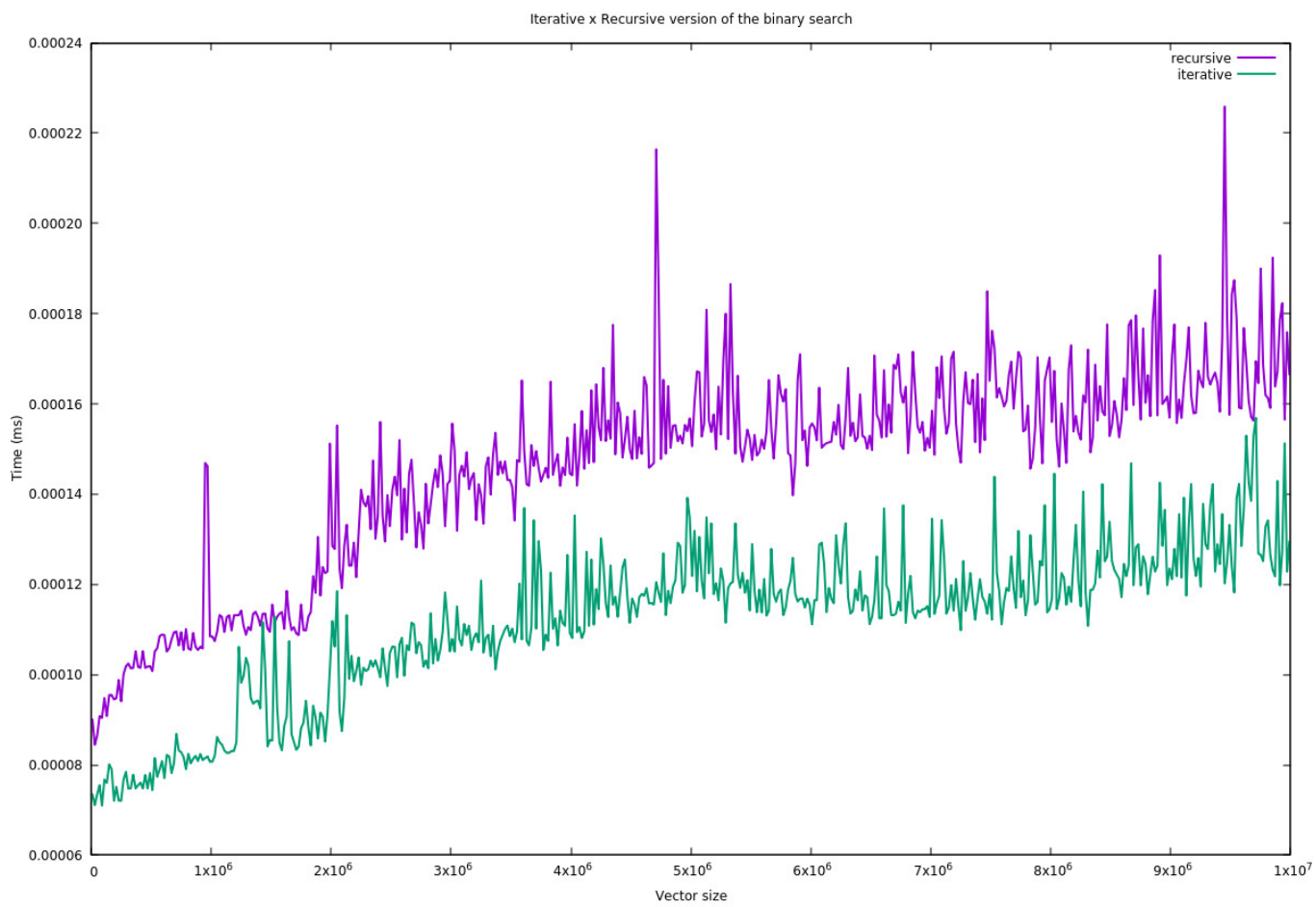


Figura 2: Gr fico referente ao item 2.