

INSTITUTO FEDERAL DE SÃO PAULO
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

FERNANDA MARTINS DA SILVA

BUSCA BINÁRIA COM PARADIGMA DE DIVISÃO E CONQUISTA

Exercício 05 listex 03 - Projeto e Análise de Algoritmos

São João da Boa Vista/SP

Etapas da Busca Binária

Como mostra a FIGURA 1, utilizando o algoritmo solicitado pelo exercício, foi implementado `buscaBinariaRecursiva()`, e separado suas etapas: divisão, conquista e combinação.

A etapa de combinação ocorre implicitamente durante toda a função `buscaBinariaRecursiva()`.

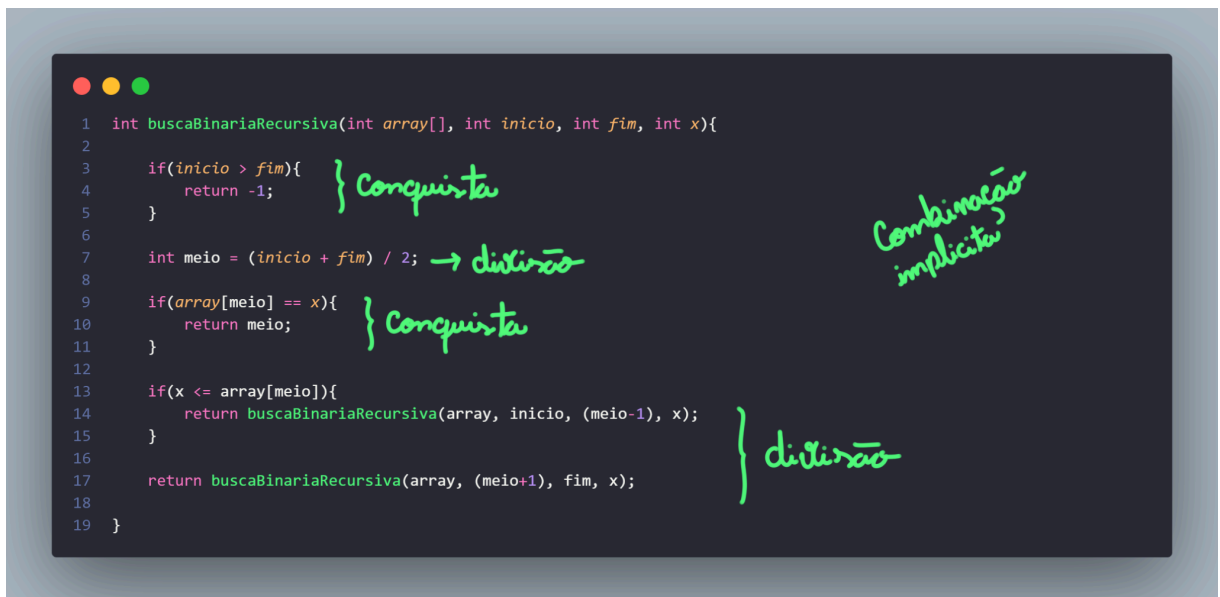


FIGURA 1

Análise de complexibilidade (T(n))

-Número de operações

Analisando a FIGURA 2, podemos perceber que, mesmo em seu melhor tempo, pior ou médio, o $T(n)$ sempre será $\log n$, isso porque a cada iteração, o tamanho do espaço de busca é reduzido pela metade, e mesmo que o elemento desejado esteja na primeira comparação (no meio do *array*) ainda precisamos realizar uma operação para fazer a comparação e verificar se encontramos o elemento desejado. Isto se deve também ao fato de que a busca binária recursiva utiliza uma quantidade constante de espaço adicional para a recursão, já que o número de chamadas recursivas é proporcional à altura da árvore de recursão.

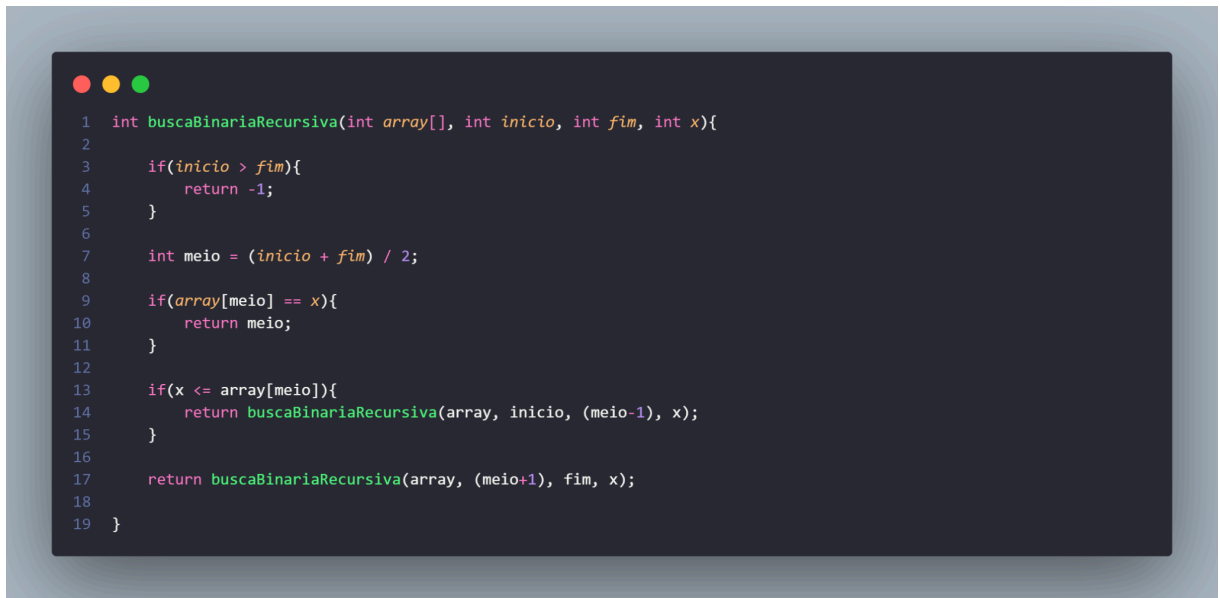


FIGURA 2

-Fórmula de recorrência

A fórmula de recorrência para o algoritmo de busca binária é

$$T(n) = T\left(\frac{n}{2}\right) + O(1)$$

onde $T(n/2)$ representa o número de operações necessárias para resolver um problema de metade do tamanho original, pois estamos dividindo o problema em duas partes iguais a cada passo. E $O(1)$ representa o tempo constante necessário para cálculo do índice médio.

-Árvore

A árvore de recursão para a busca binária pode ser representada como mostra a FIGURA 3, onde cada nível representa uma divisão do conjunto de dados pela metade.

Cada nível da árvore tem $O(1)$ operações, o número total de níveis é $\log n$, onde n é o tamanho do *array*. A árvore demonstra que a busca binária recursiva é altamente eficiente, especialmente para conjuntos de dados grandes.

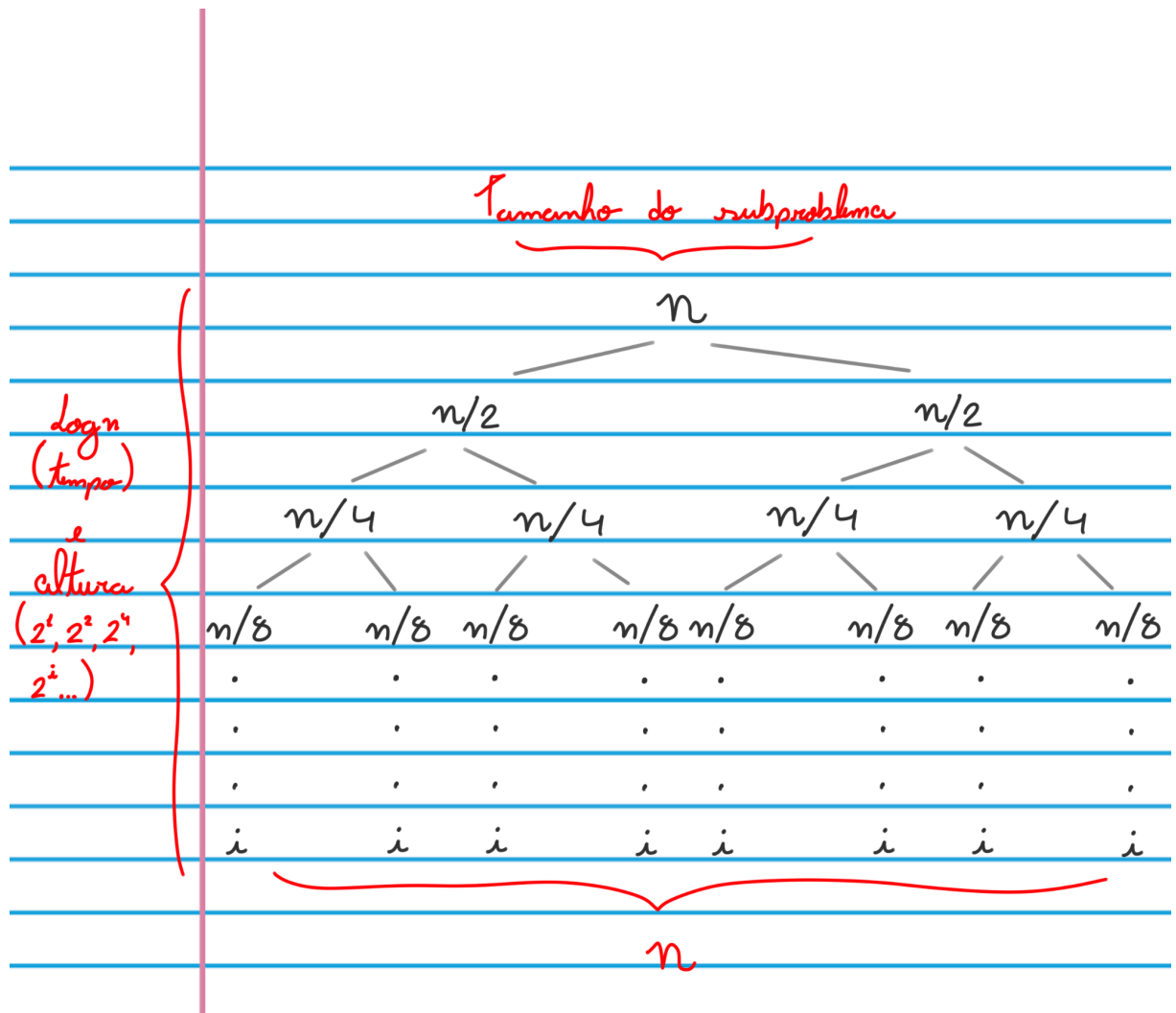


FIGURA 3