

Note: The other Tasks were all done in the Files Main.cpp, Functions.h and Matrix.h. Also all Questions were answered directly in the implementation using comments.

1 Task 7 - complexity analysis

Given the complexity analysis of Binary Search from the lecture:

Assumption: elements in the array are sorted

$c = 8$ $S = 1 \ 4 \ 7 \ 8 \ 10 \ 15 \ 17 \ 19 \ 20$ $n = 9$

Diagram illustrating the binary search process on the array S . The search range is initially $[1, 9]$. The middle element is 10 . Since $10 > 8$, the search range is updated to $[1, 8]$. The next middle element is 7 . Since $7 < 8$, the search range is updated to $[8, 8]$. The final element is 8 , which is the target. The process is labeled "found".

contains $H(\text{left}, \text{right}, c)$

Runtime analysis:

$T_{\text{bot}}(n) \in \mathcal{O}(1)$
 $T_{\text{worst}}(n) \in \mathbb{N}$
 Assumption: $n = 2^k - 1$ $k \in \mathbb{N}_{\geq 1}$
 $T(1) = a$
 $T(n) = b + T\left(\frac{n+1}{2} - 1\right) = b + b + T\left(\frac{n+1}{4} - 1\right) = \dots = \underbrace{b + \dots + b}_{k \text{ steps}} + T(1) = b \cdot \log_2(n) + a \in \mathcal{O}(\log_2(n))$

The Binary search thus has a complexity of $\mathcal{O}(\log_2(n))$ which in term is $\mathcal{O}(\log(n))$.

For the complexity analysis of the Ternary search consider:

ordered elements

$c = 8$ $S = 1 \ 4 \ 7 \ 8 \ 10 \ 15 \ 17 \ 19 \ 20$ $n = 9$ $[1, 9]$
 $[4, 5]$

Diagram illustrating the ternary search process on the array S . The search range is initially $[1, 9]$. The first split point is $m_1 = \frac{2 \cdot 1 + 9}{3} = 3$, and the second split point is $m_2 = \frac{2 \cdot 9 + 1}{3} = 6$. The middle element is 8 . Since $8 > 8$, the search range is updated to $[4, 5]$. The next middle element is 10 . Since $10 > 8$, the search range is updated to $[4, 5]$. The final element is 8 , which is the target. The process is labeled "found".

Assumption: $n = 3^k$ $k \in \mathbb{N}_{\geq 1}$
 $T(1) = a$
 $T(n) = b + T\left(\frac{n}{3}\right) = b + b + T\left(\frac{n}{9}\right) = \dots = \underbrace{b + \dots + b}_{k \text{ steps}} + T(1) = \log_3(n) \cdot b + a \in \mathcal{O}(\log_3(n))$

The Ternary search thus has a complexity of $\mathcal{O}(\log_3(n))$ which in term is also $\mathcal{O}(\log(n))$

We conclude, that both concepts have the same order of complexity. Although the Binary search needs less comparisons, which makes it the more efficient variant in most cases.