Tecnológico de Monterrey - Campus Monterrey
School of Engineering and Sciences
Department of Engineering and Sciences
Programming of Data Structures and Fundamental Algorithms

# The Use of BST in Network Infection Detection

Group #606

Santiago Quintana Moreno A01571222

Head teacher:
Luis Maltos

Monterrey, Nuevo León, 8th of September 2024

Importance of Binary Search Trees (BST)

Binary Search Trees (BST) are a crucial data structure for managing and querying ordered datasets efficiently. Their structure allows for average-case $(O(\log n)$ time complexity for search, insertion, and deletion, making them ideal for scenarios involving large, dynamic datasets. In detecting network infections, BSTs provide a mechanism for organizing and retrieving data patterns, such as malicious signatures or IP addresses, with high efficiency.

Efficiency in Problem Scenarios

In a network context, infection detection often involves analyzing logs, packet data, or file hashes to identify anomalies. Using a BST to store known infection patterns (e.g., malware signatures, blacklisted IPs) facilitates rapid comparisons. For example, as network packets arrive, their properties can be checked against entries in the BST. If the network patterns match any stored infection markers, the system can quickly flag the network as compromised.

Compared to linear structures, which require $(O(n)$ search time, BSTs significantly reduce latency in detecting threats, especially in large-scale networks. When implemented as balanced variants (e.g., AVL or Red-Black trees), BSTs maintain consistent $(O(\log n)$ performance, crucial for real-time detection systems.

Reflective Analysis

While BSTs offer efficient operations, their effectiveness depends on balanced trees. Unbalanced BSTs degrade to $(O(n))$ in worst-case scenarios, potentially slowing down detection. Moreover, hash-based structures or tries might outperform BSTs in some specific applications, such as when exact matches or prefix matching are required.

Conclusion

BSTs provide a robust and efficient solution for organizing and querying infection patterns in network detection systems. However, their use should be evaluated against the specific requirements of the problem, ensuring the structure aligns with the scale and nature of network data for optimal results.