# Empirical Analysis:

**Time Complexities:**

Binary Tree Population: O(n)

Memory Request using Preorder Traversal: O(n)

Tree Traversal to find the optimum node: O(n)

Memory Allocation: O(n)

Memory Release using Postorder Traversal: O(n)

Total Average Case Time Complexity: O(n)

**Real time data analysis (2.7 GHz i7, 16GB RAM):**

**Test Data:** Height of the tree is 10.

Computation time based on number of requests, without releasing memory:

|  |  |
| --- | --- |
| Requests | Time Taken |
| 1000 | 0 |
| 10000 | 0 |
| 50000 | 0 |
| 100000 | 1 |
| 500000 | 6 |
| 1000000 | 11 |
| 2000000 | 22 |
| 5000000 | 63 |
| 10000000 | 112 |
| 20000000 | 262 |

Computation Time with memory release:

|  |  |
| --- | --- |
| Requests | Time Taken |
| 1000 | 0 |
| 10000 | 0 |
| 50000 | 0 |
| 100000 | 0 |
| 500000 | 2 |
| 1000000 | 4 |
| 2000000 | 9 |
| 5000000 | 23 |
| 10000000 | 42 |
| 20000000 | 94 |

Computation Time : For special case by picking nodes of both far ends of trees with memory release

|  |  |
| --- | --- |
| Requests | Time Taken |
| 1000 | 0 |
| 10000 | 0 |
| 50000 | 0 |
| 100000 | 0 |
| 500000 | 2 |
| 1000000 | 3 |
| 2000000 | 6 |
| 5000000 | 16 |
| 10000000 | 32 |
| 20000000 | 74 |

Computation Time : For special case by picking nodes of both far ends of trees without memory release

|  |  |
| --- | --- |
| Requests | Time Taken |
| 1000 | 0 |
| 10000 | 0 |
| 50000 | 0 |
| 100000 | 0 |
| 500000 | 2 |
| 1000000 | 3 |
| 2000000 | 6 |
| 5000000 | 19 |
| 10000000 | 60 |
| 20000000 | 114 |