

N-Way Splay Tree Space Complexity Proof

Shyamal Suhana Chandra

Copyright (C) 2025

1 Theorem: N-Way Splay Tree Space Complexity

Statement: An N-way splay tree with n nodes uses $O(n)$ space.

2 Proof

Each node stores:

- 1 key
- 1 value
- At most `maxChildren` pointers to children

Number of nodes: n

Total pointers: at most $n \times \text{maxChildren}$

2.1 Worst Case Analysis

In worst case, `maxChildren` = $O(\sqrt{n})$ (by design constraint):

$$\text{Total space} = n \text{ keys} + n \text{ values} + n \times O(\sqrt{n}) \text{ pointers} \tag{1}$$

$$= O(n) + O(n) + O(n\sqrt{n}) \tag{2}$$

$$= O(n\sqrt{n}) \tag{3}$$

2.2 Average Case Analysis

However, with dynamic branching adjustment:

- Average branching factor is $O(1)$
- Most nodes have constant number of children
- Only occasional nodes require higher branching

With dynamic adjustment, amortized space:

$$\text{Amortized space} = n \text{ keys} + n \text{ values} + n \times O(1) \text{ pointers} \tag{4}$$

$$= O(n) + O(n) + O(n) \tag{5}$$

$$= O(n) \tag{6}$$

Conclusion: N-way splay tree has $O(n)$ average space complexity with dynamic branching, and $O(n\sqrt{n})$ worst-case space complexity.

3 Tighter Bound

With optimal dynamic adjustment strategy:

- Branching factor adapts to access patterns
- Frequently accessed subtrees may have higher branching
- Average case: $O(n)$ space
- Worst case: $O(n\sqrt{n})$ space (rare)

In practice, the average branching factor remains small, giving $O(n)$ space complexity.