Abstract

We propose to maintain a Binary Search Tree in the form of a forest in such a way that – (a) it provides faster node access and, (b) it becomes more compatible with the concurrent
environment. Using a small array, the stated goals were achieved without applying any restructuring algorithm. Empirically, we have shown that the proposed method brings down the total internal path-length of a Binary Search Tree quite considerably. The experiments were conducted by creating two different data structures using the same input - a conventional binary search tree, and a forest of hashed trees. Our empirical results suggest that the forest so produced has lesser internal path length and height in comparison to the conventional tree. A binary search tree is not a well-suited data structure for concurrent processing. The evidence also shows that maintaining a large tree in form of multiple smaller trees (forest) increases the degree of parallelism.

Reference


**Index Terms**

Computer Science  
Algorithms

**Key words**

Binary Search Tree Path Length  
Parallel Processing Binary Search

Tree
Balanced Tree
A Forest of Hashed Binary Search Trees with Reduced Internal Path Length and better Compatibility with the Concurrent Environment