Abstract

The binary search is a method of finding the position of an element in an ordered array. It continuously aims the middle element of array and check if it is the target element or not until it finds its position. The best case complexity of binary search is $O(1)$, whereas average and worst case time complexity is $O(\log n)$, where 'n' is the number of elements in the array. In this paper, I have proposed an algorithm which drastically improves the complexity of search algorithm in sorted array domain outperforming binary search, the paper also compares the proposed solution with other well-known search algorithms. The presented approach minimizes the space complexity, eliminates the need to analyze the scenario and look for the algorithm that best fits the given problem. The proposed algorithm has constant space complexity ($O(1)$) and time complexity of $O(1)$ (constant time) in best case, $O(\log(\log n))$ in average case and $O(\log(n))$ in worst case. Thus, most of the time proposed algorithm works very well as compared to other search algorithms in sorted array domain.

References
ASH Search: Binary Search Optimization


Index Terms

Computer Science

Algorithms
Keywords

Binary search, Sorted searching, Search optimization, Interpolation search, complexity analysis.