2022 Developer Survey is open! Take survey.

## Time complexity of binary search for an unsorted array

Asked 9 years, 1 month ago Modified 3 years, 4 months ago Viewed 48k times



I am stuck up with two time complexities. To do a binary search with sorted array is O(logN). So to search an unsorted array we have to sort it first so that becomes O(NlogN). So then we can perform binary search which gives the complexity as O(N) but I have read that it could be O(NlogN). Which is correct?

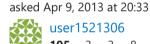


arrays sorting binary-search



11

Share Edit Follow



It's true, a binary search will not work on an unsorted array; however, you'd first have to sort the array to perform the binary search. At least that's how I have learned recently. – user2373448 Nov 21, 2017 at 21:16

## 3 Answers

Sorted by: Highest score (default)



Binary Search is for "Sorted" lists. The complexity is O(logn).

Binary Search does not work for "un-Sorted" lists. For these lists just do a straight search starting from the first element; this gives a complexity of O(n). If you were to sort the array with MergeSort or any other O(nlogn) algorithm then the complexity would be O(nlogn).



O(logn) < O(n) < O(nlogn)



Share Edit Follow

edited Mar 19, 2015 at 22:43

answered Apr 9, 2013 at 21:31



1496 2 15 2





The answer to your question is in your question itself.



You are first sorting the list. If you sort your list using quick or merge sort, the complexity becomes o(n\*log n). Part - 1 gets over.



Second part of performing a binary search is done on the 'Sorted list'. The complexity of binary search is  $o(\log n)$ . Therefore ultimately the complexity of the program remains  $o(n*\log n)$ .



However, if you wish to calculate the median of the array, you don't have to sort the list. A simple application of a linear, or sequential, search can help you with that.

Share Edit Follow

edited Aug 21, 2018 at 10:33



Olly

**7,269** 1 19 3

answered Dec 2, 2013 at 18:19



yuvrajm

6 1 3 1



The time complexity of linear search is O(n) and that of binary search is  $O(\log n)$  (log base-2). If we have an unsorted array and want to use binary search for this, we have to sort the array first. And here we have to spend a time  $O(n \log n)$  to sort the array and then spend time to search element.



Share Edit Follow



edited Nov 19, 2016 at 7:41



skrtbhtngr

**,184** 21 29

answered Nov 19, 2016 at 6:37

