

Re: ArrayList BinarySearch vs Contains

Source:

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.languages.csharp/2006-11/msg02137.html>

- *From:* "Bill Butler" <qwerty@xxxxxxx>
 - *Date:* Sun, 12 Nov 2006 03:02:11 GMT
-

"Dave Sexton" <dave@jwa[remove.this]online.com> wrote in message
<news:u%23TvmDdBHHA.3620@xxxxxxxxxxxxxxxxxxxxxxxxxxxx>

Hi Bill,

It seems you are suggesting that the scale of the data can affect whether a required sorting operation will actually reduce the overall performance of an algorithm so that it becomes worse than a linear search. Can this be true even when the data is sorted as it's being entered into the structure?

Absolutely.

Search algorithms have two distinct pieces to them

- Comparisons : Is this the right element? It is greater than this? Less than this?
- What is the next element in the collection that I should compare against.

A Linear search does a Fast Compare and a fast iteration to the next element, but it doesn't apply any logic to reduce the number of comparisons required. Therefore, for small data sets, a linear search can be the best option. For large sets, you pay the price of having to compare against each and every element until you find the correct one.

A Binary search against a sorted dataset has a fairly fast compare with a slower iteration to the next element, since it needs to calculate where it should do the next compare. This additional logic in the algorithm means that each step in the algorithm is slower than the linear algorithm, but it drastically reduces the number of steps involved. Therefore the Binary search is faster when the number of elements becomes the dominant factor.

The original question asked about the best way to find elements in a sorted ArrayList, but I would like to bring up an alternative. Try using a Hashtable instead. It has a higher memory footprint, but the performance is exceptional. For completeness in this discussion I would

Re: ArrayList BinarySearch vs Contains

like to point out that Hashtable retrieval is a constant time operation $O(1)$. I ran a quick test comparing Hashtable vs Linear search vs BinarySearch and the results are impressive. Hashtable wins hands down when more than a handful of elements are involved

Bill