

Fundamentals of Algorithm

SUNAWAR KHAN
MSCS
IIUI

Linear Search

- ▶ The linear search is a sequential search, which uses a loop to step through an array, starting with the first element.
- ▶ It compares each element with the value being searched for, and stops when either the value is found or the end of the array is encountered.
- ▶ If the value being searched is not in the array, the algorithm will unsuccessfully search to the end of the array.

Linear Search

- ▶ Since the array elements are stored in linear order searching the element in the linear order make it easy and efficient.
- ▶ The search may be successful or unsuccessfully. That is, if the required element is found then the search is successful otherwise it is unsuccessful.

Advantages

- ▶ The linear search is simple - It is very easy to understand and implement
- ▶ It does not require the data in the array to be stored in any particular order

Disadvantages

- ▶ It has very poor efficiency because it takes lots of comparisons to find a particular record in big files
- ▶ The performance of the algorithm scales linearly with the size of the input
- ▶ Linear search is slower than other searching algorithms

Linear Search Example

-23	97	18	21	5	-86	64	0	-37
-----	----	----	----	---	-----	----	---	-----

↑
element

Searching for -86.

Linear Search Example

-23	97	18	21	5	-86	64	0	-37
-----	----	----	----	---	-----	----	---	-----

↑
element

Searching for -86.

Linear Search Example

-23	97	18	21	5	-86	64	0	-37
-----	----	----	----	---	-----	----	---	-----

↑
element

Searching for -86.

Linear Search Example

-23	97	18	21	5	-86	64	0	-37
-----	----	----	----	---	-----	----	---	-----

↑
element

Searching for -86.

Linear Search Example

-23	97	18	21	5	-86	64	0	-37
-----	----	----	----	---	-----	----	---	-----

↑
element

Searching for -86.

Linear Search Example

-23	97	18	21	5	-86	64	0	-37
-----	----	----	----	---	-----	----	---	-----

↑
element

Searching for -86: found!

Analysis of Linear Search

How long will our search take?

In the best case, the target value is in the first element of the array.

So the search takes some tiny, and constant, amount of time.

In the worst case, the target value is in the last element of the array.

So the search takes an amount of time proportional to the length of the array.

Analysis of Linear Search

In the average case, the target value is somewhere in the array.

In fact, since the target value can be anywhere in the array, any element of the array is equally likely.

So on average, the target value will be in the middle of the array.

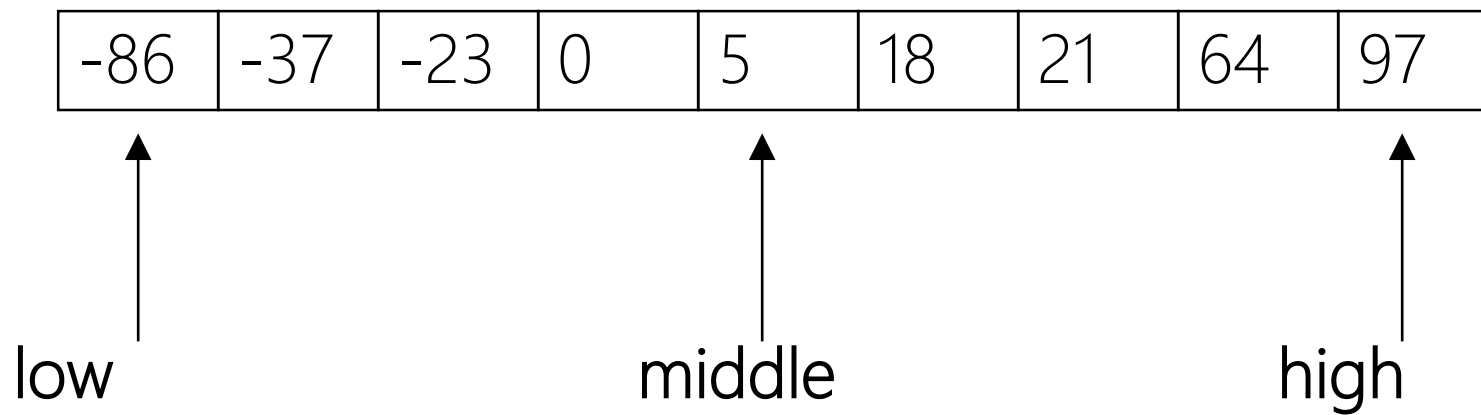
So the search takes an amount of time proportional to half the length of the array

Binary Search

The general term for a smart search through sorted data is a binary search.

1. The initial search region is the whole array.
2. Look at the data value in the middle of the search region.
3. If you've found your target, stop.
4. If your target is less than the middle data value, the new search region is the lower half of the data.
5. If your target is greater than the middle data value, the new search region is the higher half of the data.
6. Continue from Step 2.

Binary Search Example



Searching for 18.

Binary Search Example

-86	-37	-23	0	5	18	21	64	97
-----	-----	-----	---	---	----	----	----	----

low middle high

The diagram illustrates the initial state of a binary search on a sorted array. Three arrows point upwards from the labels 'low', 'middle', and 'high' to the element '18' in the array. 'low' points to the 6th element (index 5), 'middle' points to the 7th element (index 6), and 'high' points to the 9th element (index 8). The array is: [-86, -37, -23, 0, 5, 18, 21, 64, 97].

Searching for 18.

Binary Search Example

-86	-37	-23	0	5	18	21	64	97
-----	-----	-----	---	---	----	----	----	----

low high
 ↑ ↑
 middle

Searching for 18: found!

Time Complexity of Binary Search

How fast is binary search?

Think about how it operates: after you examine a value, you cut the search region in half.

So, the first iteration of the loop, your search region is the whole array.

The second iteration, it's half the array.

The third iteration, it's a quarter of the array.

...

The k^{th} iteration, it's $(1/2^{k-1})$ of the array.

Need to Sort Array

Binary search only works if the array is already sorted.

It turns out that sorting is a huge issue in computing – and the subject of our next lecture.