Searching Algorithms

Linear Search

Linear Search

- The Lineary Search algorithm allows the user to search for a specific value in a list of values.

- The searching starts at the beginning.

- Stops when the item is found OR when the search reaches the end of the list.

- The worst-cast performance: O(N)

Linear Search Challenge

1. Create a method called LinearSearch that takes a list of ints as a parameter and 1 int as the search number.

2. In the method, use the linear search algorithm to find the item.

3. Return the index of the item. Return -1 if the item is not found.
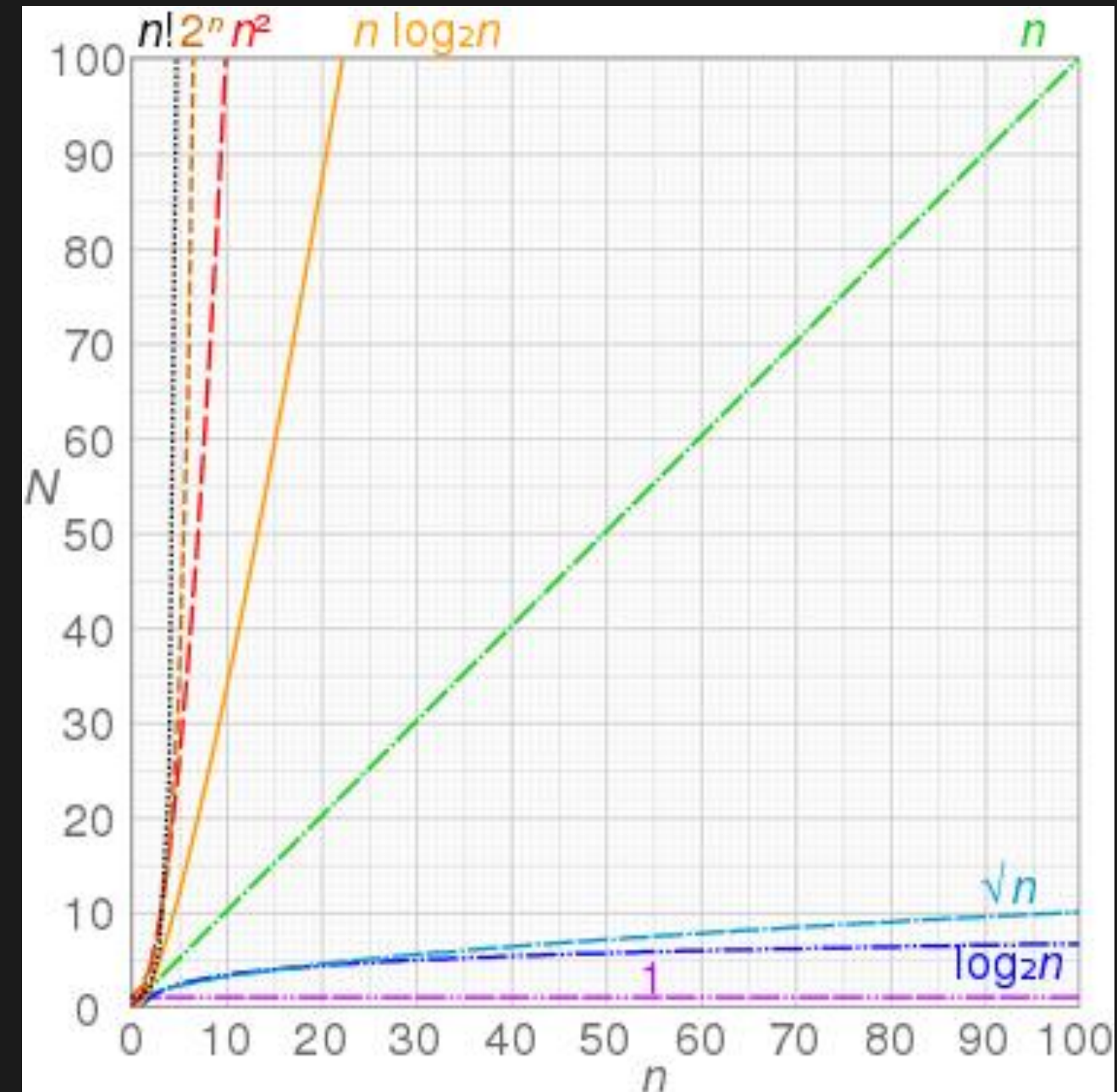
4. Call LinearSearch from Main.

Binary Search

Binary Search

- The Binary Search algorithm is a very efficient search: **O(log n)**

- **ONLY** works on sorted data

- Divides-and-conquers!

Computational Complexity

Best

O(1)	constant
O(log n)	logarithmic
O(n)	linear
O(n log n)	loglinear
O(n^2)	quadratic
O(2^n)	exponential
O(n!)	factorial

Worst

Binary Search

- The Binary Search algorithm:

  - If the middle item is your search term, quit! You found it!

    - Return the index of the middle item

  - Else if the search term is LESS than the middle item

    - repeat the search with the left half

  - Else if the search term is GREATER than the middle item

    - repeat the search with the right half

  - If the min index > max index, return -1 (this is an exit condition)

Binary Search pseudocode

```
// initially called with low = 0, high = N-1
BinarySearch(A[0..N-1], searchTerm, low, high)
{

    if (high < low)
        return -1 // -1 means not found

    mid = (low + high) / 2

    if (A[mid] > searchTerm)
        return BinarySearch(A, searchTerm, low, mid-1)

    else if (A[mid] < searchTerm)
        return BinarySearch(A, searchTerm, mid+1, high)

    else
        return mid //the searchTerm was found so return its index

}
```