Data Structures and Algorithms

Linear Search and Binary Search

Amilcar Aponte

amilcar@cct.ie

Searching

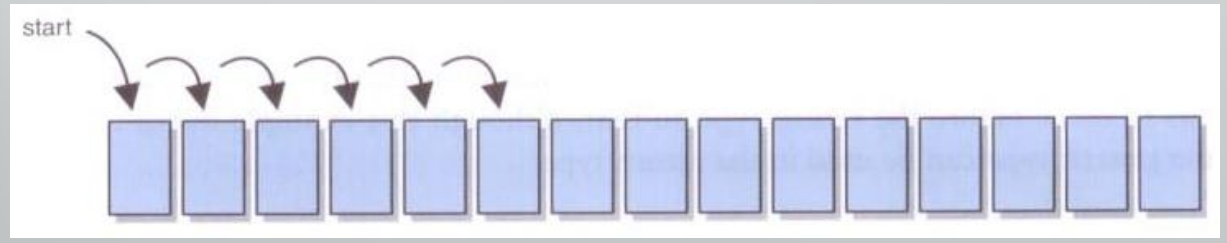
- ***Searching*** is the process of finding a *target element* among a group of items (the *search pool*) or determining that it isn't there.
- This requires repetitively comparing the target to candidates in the search pool.
- An efficient search performs no more comparisons than it has to.

Searching

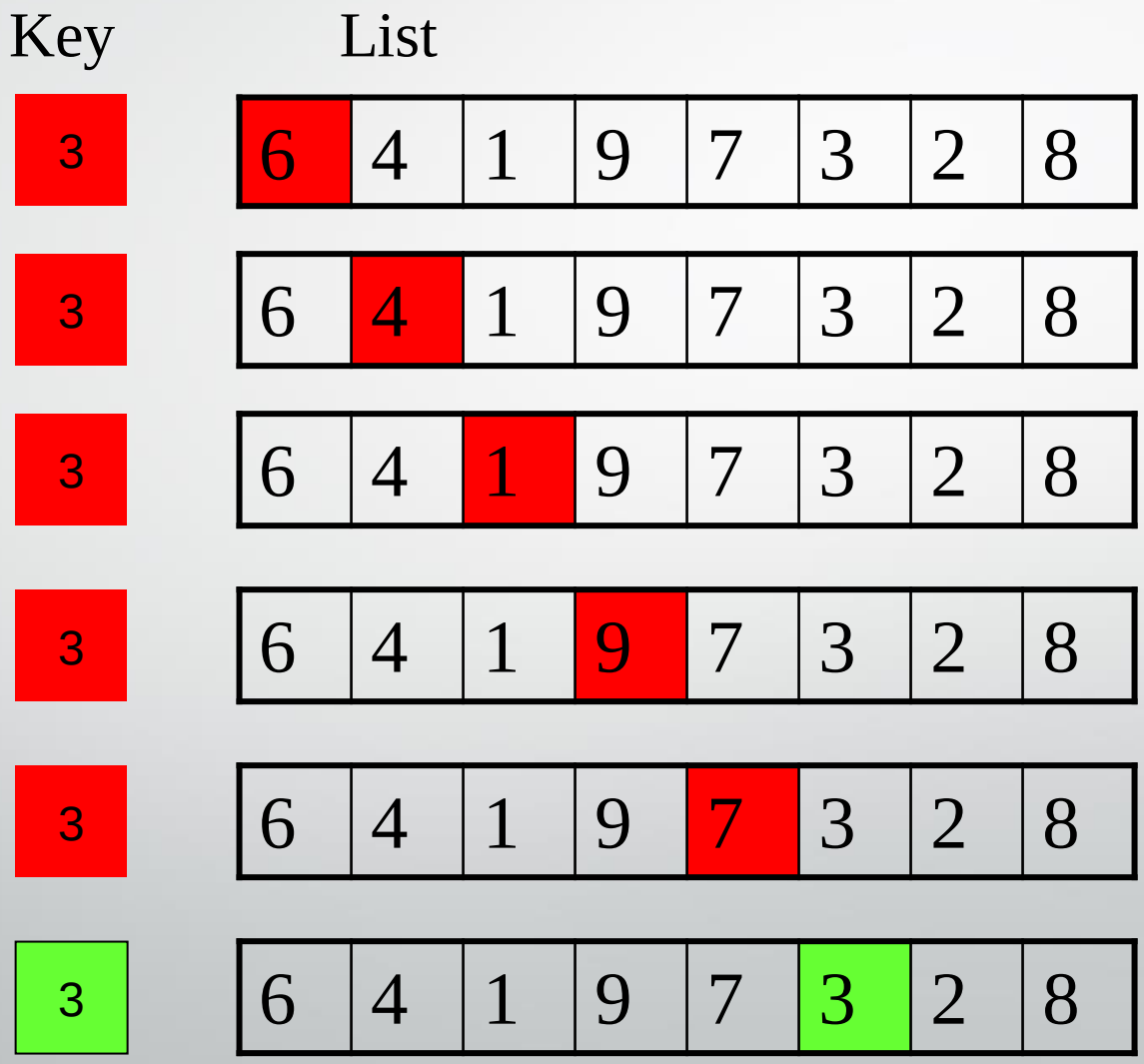
- When the element is found, my searching algorithm should return the position in which one could find the element inside the pool.
- In some more complex cases we might want to get the element it self, but that could be implemented once we understand the mechanics of any of the algorithms.

Linear Search

- The linear search approach compares the key element, key, ***sequentially*** with each element in the list.
- The method continues to do so until the key matches an element in the list, or the list is exhausted without a match being found.
- If a match is made, the linear search returns the index of the element in the array that matches the key. If no match is found, the search returns -1.



Linear Search Animation



https://liveexample.pearsoncmg.com/dsanimation/LinearSearcheBook.html

Let’s code this

- Let’s apply this to an ArrayList.
- But this can be done in any list or array.

Binary Search

- If the search pool is sorted, then we can be more efficient than a linear search.
- A ***binary search*** eliminates large parts of the search pool with each comparison.
- Instead of starting the search at one end, we begin in the middle.
- If the target isn't found, we know that if it is in the pool at all, it is in one half or the other.
- We can then jump to the middle of that half, and continue similarly.

Binary Search

- The binarySearch method returns the index of the element in the list that matches the search key if it is contained in the list. Otherwise, it returns - 1.

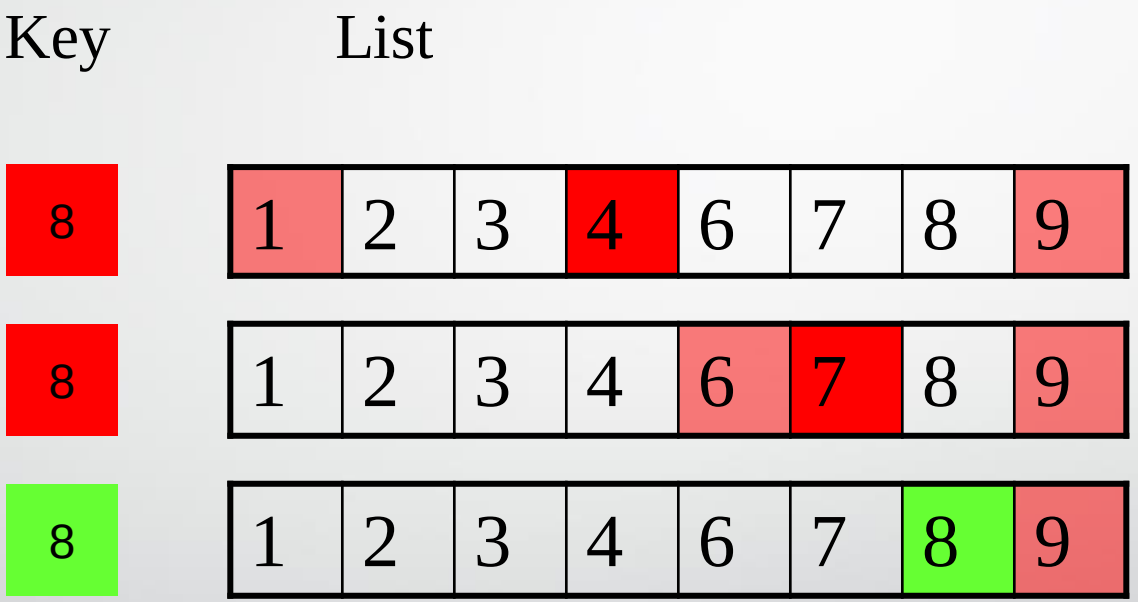
Binary Search

- For binary search to work, the elements in the array must already be ordered. Without loss of generality, assume that the array is in ascending order.
  - e.g., 2 4 7 10 11 45 50 59 60 66 69 70 79
- The binary search first compares the key with the element in the middle of the array.

Binary Search

- Consider the following three cases:
- If the key is ***less than*** the middle element, you only need to search the key in the first half of the array.
- If the key is ***equal to*** the middle element, the search ends with a match.
- If the key is ***greater than*** the middle element, you only need to search the key in the second half of the array.

Binary Search



https://liveexample.pearsoncmg.com/dsanimation/BinarySearcheBook.html

Let’s code this

- Let’s apply this to an ArrayList.
- But this can be done in any list or array.

Take a look at this..

- https://www.mathwarehouse.com/programming/gifs.php#binary-vs-linear-search-gif

That’s all folks

- Any questions?