Summary – Searching

Linear or Sequential Search – Examines all values in an array until it find a match or reaches the end. Array does not need to be sorted. The average search visits n/2 elements, so a linear search is an O(n) algorithm. There is no faster way to do a search in an unsorted array.

Binary Search – locates a value in a sorted array by determining whether the value occurs in the first or second half, then repeating the search in one of the halves until it is found. For example, if you performed a binary search on the array {1,5,8,9,12,17,20,32} to find the value 32. The search sees that 32 is greater than 9 (the last value of the first of half of the array), so it looks only at the elements in the second half of the array {12,17,20, 32} and so on until it locates 32. A binary search is an O(log(n)) algorithm. The Arrays class contains a static binarySearch method that implements the binary search algorithm. If a value is not found in the array then the return value is –k-1, where k is the position before which the element should be inserted. For example, if you wanted to insert 5 in the array {2,3,7,8,11}, -3 would be returned since 5 should be inserted before position 2.

A binary search is much faster, but you need a sorted array which is an o(log(n)*n) algorithm. As a rule of thumb, you should use linear searches when you do not have to make many searches, and when you need to make many searches, you should pay o(n*log(n)) for sorting the array, and then use binary search.

Searches can be called either recursively or iteratively.