Interpolation Search

Linear Search finds the element in O(n) time, Jump Search takes O(√n) time and Binary Search take O(Log n) time.
The Interpolation Search is an improvement over Binary Search for instances, where the values in a sorted array are uniformly distributed. Binary Search always goes to the middle element to check. On the other hand, interpolation search may go to different locations according to the value of the key being searched. For example, if the value of the key is closer to the last element, interpolation search is likely to start search toward the end side.

Linear interpolation

Generally, linear interpolation takes two data points, say (x_a, y_a) and (x_b, y_b), and the interpolant is given by:

$$y = y_a + (y_b - y_a)\,\frac{x - x_a}{x_b - x_a} \text{ at the point } (x, y)$$

$$\frac{y - y_a}{y_b - y_a} = \frac{x - x_a}{x_b - x_a}$$

$$\frac{y - y_a}{x - x_a} = \frac{y_b - y_a}{x_b - x_a}$$

This previous equation states that the slope of the new line between (x_a, y_a) and (x, y) is the same as the slope of the line between (x_a, y_a) and (x_b, y_b)

Linear interpolation is quick and easy, but it is not very precise. Another disadvantage is that the interpolant is not differentiable at the point x_k.

The following error estimate shows that linear interpolation is not very precise. Denote the function which we want to interpolate by g, and suppose that x lies between x_a and x_b and that g is twice continuously differentiable. Then the linear interpolation error is

$$|f(x) - g(x)| \le C(x_b - x_a)^2 \quad \text{where} \quad C = \frac{1}{8} \max_{r \in [x_a, x_b]} |g''(r)|.$$

In words, the error is proportional to the square of the distance between the data points. The error in some other methods, including polynomial interpolation and spline interpolation (described below), is proportional to higher powers of the distance between the data points. These methods also produce smoother interpolants.

To find the position to be searched, it uses following formula

```
position = lo + [(x - arr[lo])*(hi - lo) / (arr[hi] - arr[Lo])]

arr[] - Array where elements need to be searched
x     - Element to be searched
lo    - Starting index in arr[]
hi    - Ending index in arr[]
```

Algorithm

Interpolation Search algorithm is same to binary search except the partition logic.

Step1: In a loop, calculate the value of "position" using the probe position formula.
Step2: If it is a match, return the index of the item, and exit.
Step3: If the item is less than arr[position], calculate the probe position of the left sub-array. Otherwise calculate the same in the right sub-array.
Step4: Repeat until a match is found or the sub-array reduces to zero.

If the elements are uniformly distributed, then time complexity is O (log log n)), its proof is complex.
In worst case it can take up to *O*(n).