Sandeep Heera

Interpolation Search Example

**Algorithm** InterpolationSearch(A[1…*n*], key)

**Input:** a set of *n* ordered integers arranged in ascending order and an integer *key* to search for

**Output**: index of the location of the key if found or -1 if it is not in the set

*l*←1

*r←n*

**if**(*n* = 0) **then**

**return** -1

**else if**(A[*l*] > *key* **or** A[*r*] < *key*) **then**

**return** -1

**else**

**while**(A[*l*] <= *key* **and** A[*r*] >= *key*) **do**

**if**(A[*x*] = *key*) **then**

**return** *x*

**else if**(A[*x*] < *key*) **then**

*l←x* + 1

**else**

*r←x* – 1

**return** -1

Example

For *key* = 98, initially *l* = 1, *r* = *n* = 12, A[1] = 20, A[12] = 210, A[5] = 50, A[7] = 98.

Since 98 is in between A[*l*] and A[*r*], we will enter the while loop.

First Pass:

The algorithm will now check A[5] and compare it to the *key*. A[5] is equal to 50 which is less than the *key*, so the algorithm will make *l* equal to 5 + 1 = 6 and go through the loop again.

Second Pass:

*l =* 6 and *r* = 12

The algorithm will now check A[7] and compare it to *key*. A[7] is equal to the *key* 98 and the algorithm will return 7.