

Interpolation Search in R

Rahul Goswami

2022-01-18

Interpolation Search is a searching algorithm that is used to find an element in a sorted array. The basic idea of interpolation search is to search for an element in a sorted array by linearly interpolating between the elements.

The average case time complexity of interpolation search is $O(\log n)$

The worst case time complexity of interpolation search is $O(n)$

The best case time complexity of interpolation search is $O(\log n)$ (if the array is already sorted)

The space complexity of interpolation search is $O(1)$ (in-place)

Algorithm: (For more details, refer to the book “Algorithms” by Robert Sedgewick and Kevin Wayne)

1. Find the index of the element to be searched (i.e. the index of the element in the array that is just greater than the element to be searched)
2. If the element to be searched is less than the element at the index, search the first half of the array
3. If the element to be searched is greater than the element at the index, search the second half of the array
4. If the element to be searched is equal to the element at the index, return the index
@param vec Vector to be searched @param element Element to be searched @return Index of the element

```
interpolation.search <- function(vec, element){  
  # Find the index of the element to be searched  
  index <- floor(length(vec)*(element-vec[1])/(vec[length(vec)]-vec[1]))+1  
  cat("index: ", index)  
  # If the element to be searched is less than the element at the index, search the first half of the  
  if(vec[index] > element){  
    return(interpolation.search(vec[1:index], element))  
  }  
  # If the element to be searched is greater than the element at the index, search the second half of  
  if(vec[index] < element){  
    return(index + interpolation.search(vec[(index+1):length(vec)], element))  
  }  
  # If the element to be searched is equal to the element at the index, return the index  
  return(index)  
}
```

Example

```
sorted_vec <- c(1,2,3,3,3,4,5,6,7,8,9)  
interpolation.search(sorted_vec, 5)
```

```
## index: 6index: 1
## [1] 7
```