Que 20

Binary Search vs. Interpolation Search

Binary Search=>

2	5	12	16	23	38	56	72	78	91

Sort array with Merge Sort=>

2	5	12	16	23	38	56	72	78	91

(1) (10)

[2 5 12 16 23] [38 56 72 78 91]

(2) (6) (12) (17)

[2 5 12] [16 23] [38 56 72] [78 91]

(3) (7) (8) (13) (14) (18) (19)

[2 5] [12] [16] [23] [38 56] [72] [78] [91]

(4) (9) (15) (18)

[2] [5] [12] [16] [23] [38] [56] [72] [78] [91]

(5) (10) (16) (20)

[2 5 12] [16 23] [38 56 72] [78 91]

(11) (21)

[2 5 12 16 23] [38 56 72 78 91]

(22)

[2 5 12 16 23 38 56 72 78 91]

Binary Search of number 5 in sorted array=>

Given a sorted array A[0: n-1] and a search key

= if KEY = A[m], then return m

= if KEY < A[m], then search the left half of the array.

= if KEY > A[m], then search the right half of the array

2	5	12	16	23	38	56	72	78	91

L=0 M=4 5 is smaller than 2nd half array H= 9

2	5	12	16	23	38	56	72	78	91

L=0 M=2 H=4 search 5 in first half

2	5	12	16	23	38	56	72	78	91

L=0 M=1 H=2 search in 2nd half

2	5	12	16	23	38	56	72	78	91

L= H= 1= value=5

Big-O comparison=> Time complexity Analysis=> Master theorem=>

Merge sort=>

For merging 2 sorted array , O(n1+n2)=O(n).

Binary search=>

In an array of **n** elements,

```
T(n) = 0          if n = 1
T(n) = T(n/2) + 1     otherwise
```

- Using this recurrence relation T(n) = log(n)
- Therefore, binary search uses O(log n) time.

Interpolation Search=>

0	1	2	3	4	5	6	7	8	9
2	5	12	16	23	38	56	72	78	91

L : (x0, y0) = (0, 2)

H: (x1, y1) = (9, 91)

M: (x, y) = (x, 5)

x= x0 + (y – y0)* (x1 – x0)/(y1 – y0)

= 0 + (5 -2) * (9 - 0)/(91- 2)

= 0 + 3 * 9 /89

=27/89

=0

Big-O Comparison=>

- Interpolation search is an improved variant of binary search.
  - This search algorithm works on the probing position of the required value.
  - For this algorithm to work properly, the data collection should be in a sorted form and equally distributed.

Interpolation search=> On average: O(log (log n)) ; worst case O(n)

- The best case for Interpolation Search happens when the middle (our approximation) is the desired key.
  - This makes the best case time complexity is O(1).
- In the worst-case scenario, we will need to traverse all of the elements in the array, resulting in the O(n) time complexity.
- The good news is for the average case, the time complexity is as small as O(log log n).

Conclusion:

- **Are the two approaches tied?**
- **=> NO**
-
- **If one of the approaches is better, which one is better? why?**

- **=>I think, Binary search is better than Interpolation Search.**
- **In Binary search, comparisons is done with half of the array every single time and In Interpolation search, comparisons is done with probing position of of required value using a formula. Even though time complexity for interpolation search is small than Binary search, interpolation search may require more comparisons than Binary Search.**
- **That's why Binary search is better than interpolation search.**