Quick - Sort

Best Case

$T(0) = T(1) = 0$

$T(N) = 2T(N/2) + N$

$a = 2 ; b = 2 ; \log_2 2 = 1 ; f(n) = \Theta(N)$

$T(N) = N \log N \Rightarrow \Omega(N \log N)$

Worst Case

$T(N) = N + T(N-1)$

$\therefore T(N) = O(N^2)$

FUZZY SORT

WORST CASE

The reccurence rllation of running time

$$T(n) = 2T(n/2) + \Theta(n)$$

$a = 2 ; b = 2 ; f(n) = \Theta(n)$

$\log_2 2 = 1$

$T(n) = \Theta(n \log n)$

This is because the fuzzy sort is nearly identical to randomized quick - sort.

The worst case occurs when all the input intervals are disjoint.

∴ No overlaps will be there and time complexity will be $\Theta(n \log n)$

BEST CASE of FUZZY SORT

For this, every interval will be within middle region. So, left and right subarrays will be empty having running time as $\theta(1)$.
So, to find the region of overlap, the running time is determined by $\theta(n)$.

So, all intervals overlap at a point, then expected running time will be $\theta(n)$

CONCLUSION OF FUZZY SORT

Since, the worst case running time is expected as $\theta(n \log n)$ & best case running time is $\theta(n)$.
Therefore, this fuzzy sorting can be accomplished in less than $\theta(n \log n)$

by exploiting the property that overlapping intervals can appear in any permutation in the output.

So, fuzzy sort is better & faster than quick sort.