Project 1-Fuzzy Sort

Q1 . Fuzzy Sort Algorithm

To sort the intervals (containing overlapping and distinct values), we need to use a modified version of Quick Sort. We will need to utilize the logic of overlapping intervals and improve the running time our algorithm. There can be 3 types of intervals when we compare them to the pivot interval.

Types of intervals-

1. Smaller interval (distinct)

2. Larger interval (distinct)

3. Overlapping intervals

The overlapping intervals need not to be sorted when they are overlapping to the left or right endpoints of the pivot. But, we need to sort them if they are part of the intersection and don’t overlap with right or left endpoints of the pivot interval, i.e. are contained in the pivot interval.

Let’s consider a structure array A having left and right endpoints.

**FUZZY-SORT(A,p,r)**

If p<r

q= PARTITION(A, p, r)

FUZZY-SORT(A, p, q.left)

FUZZY-SORT(A, q.right, r)

**PARTITION (A, p, r)**

x=A[r]

for i🡨p to r

if x.left <= A[i].right AND A[i].left <= x.right

if A[i].left > x.left

x.left 🡨 A[i].left

if(A[i].right < x.right)

x.right 🡨 A[i].right

s 🡨 p-1

for i<-p to r

if A[i].right < x.left

s++

EXCHANGE A[i] <---> A[s]

EXCHANGE A[r] <--> A[s+1]

t 🡨 s+1

i 🡨r

while t <= i

if x.left<=A[i].right AND A[i].left<= x.right

EXCHANGE A[i] <--> A[t]

t++

else

i—

X.right 🡨t

X.left 🡨 s

RETURN X

Q2. Running Time of Fuzzy Sort

Let’s consider the running time of the PARTITION (A, p, r) given above-

1. The first and second for loops executes the steps (r-p) -times, i.e. n-times each. So, it will be of Θ(n) order.
2. The last while loop executes the steps for (r-s+1) times, which is also linear.
3. Overall, the running time of the PARTITION(A, p, r) will be linear, i.e. Θ(n) order.

When we analyze the FUZZY-SORT(A, p, r), it works similar to Quick Sort. So, considering average performance of Quick Sort, the expected time of Fuzzy Sort algorithm will also be Θ(nlgn) in general when it sorts distinct intervals.

However, the algorithm’s running time will improve when the intervals are intersecting to the pivot. In the case where all intervals are overlapping, only the PARTITION (A, p, r) will be executed. The running time would then become Θ(n).

Q3. Graph for Number of Inputs n Vs Running Time

From the below line graphs and running times, it is clear that the values having more overlap can be sorted faster. Also, from the below line graphs this is clear that the overlapping inputs have linear running time and the small overlapping inputs have logarithmic graph.

|  |  |  |  |
| --- | --- | --- | --- |
| Number of Input | Highest Overlap | Large\_Overlap | Small\_Overlap |
| 100 | 0 | 0 | 0 |
| 500 | 0 | 0 | 0 |
| 1000 | 0 | 0 | 0.001 |
| 2000 | 0 | 0 | 0 |
| 5000 | 0 | 0 | 0.001 |
| 10000 | 0 | 0.001 | 0.003 |
| 20000 | 0 | 0.002 | 0.007 |
| 50000 | 0.001 | 0.006 | 0.019 |
| 100000 | 0.002 | 0.008 | 0.047 |
| 200000 | 0.005 | 0.025 | 0.094 |
| 500000 | 0.01 | 0.062 | 0.26 |
| 1000000 | 0.023 | 0.135 | 0.48 |
| 2000000 | 0.049 | 0.181 | 0.477 |
| 5000000 | 0.111 | 0.217 | 0.557 |
| 10000000 | 0.222 | 0.321 | 0.637 |