Assignment - 3

I choose the divide & conquer method to solve the given problem.

According to the given question, the 2 databases have same number of elements → 'n'.

From the example explanation given, the 2 databases arrays are sorted arrays.

Initially, I define a function to calculate the Median with the given Array and the ArraySize.

Defining Median function:

Median (Array, Size):

if $\frac{Size}{2} == 0$:

return $\left(\frac{Array\left[\frac{Size}{2}\right] + Array\left[\frac{Size}{2}+1\right]}{2} \right)$

else:

return $\left(Array\left[\frac{Size}{2}\right] \right)$

I define this function to call it later whenever useful.

if size == 1 :

return (min(DatabaseA[0], DatabaseB[0]))

else :

medianA = Median(DatabaseA, n)

medianB = Median(DatabaseB, n)

if medianA > medianB :

if n/2 = = 0 :

return (Recursively apply the same steps until a single element is present by dividing the arrays. All the elements until (median DatabaseA[ ]) and all the elements from median (DatabaseB[ ] till the last element)

if n/2 = = 1 :

return (Recursively apply the same steps until a single element is present by dividing the arrays. All the elements until median (DatabaseA) and all the elements from median (DatabaseB) till the last elements)

elif medianA = medianB :

return (medianA)

else :

if n/2 = = 0 :

return (Recursively apply the same steps until a single element is present by dividing the arrays, All the elements until median (DatabaseB) and all

the elements from
median (Database B) till the
last elements present)

if $n/2 == 1$;
return (Recursively apply the same steps
until a single element is present by
dividing the arrays. All the elements
until median (Database B) and all the
elements from median (Database A)
till the last elements present).

I recursively repeat the same steps by
comparing the medians of database A & database B
and further divide the arrays until I get an
array with a single element in it by using
the Median function defined above.

And among the single elements present in both
the databases the minimum value element would
be the Median value as defined in the question

Runtime Analysis :-

Initially the total number of elements present in the two databases are } $\Rightarrow$ $n+n = 2n$

Once, we divide the database array from 'n' elements into

n

$\frac{n}{2}$ $\frac{n}{2}$ 2 $(n/2)$ arrays

$\frac{n}{4}$ $\frac{n}{4}$ further divide the $\left(\frac{n}{2}\right)$ elements array into 2 $(n/4)$ elements array.

We repeat this step recursively until we get an array with single element in 2 ~~elements~~ arrays

- The runtime for the Median () function calls would be $O(1)$.

- The runtime for the comparision of medianA & medianB would be $O(1)$.

- As we know all the comparisions made also take runtime of $O(1)$

Therefore, we define $T(2n)$

~~for~~ recursively calling for $2n$ elements.

$$T(2n) = \begin{cases} T(n) + c_1 & n > 1 \\ c_2 & n = 1 \end{cases}$$

This would be the recursive recurrence for the runtime of the algorithm.

Further simplifying, $T(2n)$

$$T(n) = \begin{cases} T(n/2) + c_1 & ; n > 2 \\ c_2 & ; n = 2 \end{cases}$$

From the Master's Theorem discussed in the class:

We define

$$T(n) = aT(n/b) + n^d$$

$$T(n) = \begin{cases} O(n^{\log_b a}) & ; \text{ if } \log_b a > d \\ O(n^d) & ; \text{ if } \log_b a < d \\ O(n^d \log n) & ; \text{ if } \log_b a = d \end{cases}$$

where, a - number of sub problems
$a \geq 1$

b - size of the sub problem.

$d \geq 0$

From comparison.

$$T(n) = \begin{cases} T(n/2) + c_1 & ; \; n > 2 \\ c_2 & ; \; n = 2 \end{cases}$$

From Master's theorem,

$a = 1$; $b = 2$; $d = 0$.

Now, $\log_b a = \log_2 1 = \underline{\underline{0}}$

$\underline{\underline{d = 0}}$

$\therefore$ $\boxed{\log_b a = d}$

Hence, $T(n) = O(n^d \log n)$ if $\log_b a = d$

$\therefore$ as $d = 0$

$$\boxed{T(n) = O(\log n)}$$