7. Algorithm for finding median of 2 databases

Let A and B be 2 databases.

A[i] -> To retrieve i^{th} element in the database A

B[i] -> To retrieve i^{th} element in the database B

```
function find_median(A,a_start,a_end,B,b_start,b_end,n){
A_mid=(a_end-a_start)/2;
B_mid=(b_end-b_start)/2;
if(n==1)
return min(A_mid,B_mid);
if(A[A_mid]<B[B_mid]){
a_start=A_mid+1;
b_end=B_mid;                                    // finding mid of A and B
n=n-A_mid;
find_median(A,a_start,a_end,B,b_start,b_end,n);  //recursive call
}
else{
b_start=B_mid+1;
a_end=A_mid;
n=n-B_mid;
}
}
```

Analysis of algorithm

In each recursive call, we eliminate one half of both the arrays and consider only the other half in the next level of recursion. Thus we reduce the size of array considered by half in each level.

There is just one key comparison done in every recursive call : if(A[A_mid]<B[B_mid])

The recursion ends when either of the array has only one element. T(1)=1 (min(A_mid,B_mid))

Therefore we arrive at the recursive relation:

$T(n)=T(n/2) + 1$

applying master theorem for this, we get

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

$b^d = a$

This comes under case 2 : $T(n) \in \Theta(n^0 \log n) \in \Theta(\log n)$

Find median algorithm

1. if number of elements is 1, then minimum of the two elements is the median
2. if either A or B has only one element in it then median is minimum of first element of both the tables
3. if mid of A is lesser, it means the median lies in latter half of A or first half of B.
4. if mid of B is lesser, it means the median lies in latter half of B or first half of A.
5. repeat step1 to 4 till median is found with new values of A, B - each time eliminating 1 half of both the arrays.