Davie Truong

“I have read and agree to the collaboration policy. Davie Truong”

Homework Heavy

Collaborators: Yona Edell

**Homework 1 Question 4 (Divide and Conquer)**

A: 2 4 6 9 10 🡪 A: 6 9 10 🡪 A: 6 9

6-> <-9

B: 5 7 8 10 12 🡪 B: 5 7 8 🡪 B: 7 8

<-8 7->

**Algorithm/ Description**:

Initially find the median of both arrays. If the arrays are even, use the floor and ceiling to discover the median and then proceed by comparing the discovered medians. Compare the two medians to determine which part of the array needs to be kept. From the example above, since 6 < 8 we want the elements greater than 6 from array A and elements less than 8 from array B. The median of the two values would only be within that range. Doing this also cuts the search time by 2, leading towards the desired logn runtime. Once again, we find the median of the two arrays and compare values, continuing recursively. When each array only has 2 elements get the greater value from index 0 after comparison and the lower value from index 1 after comparison. Either of the 2 final values could be the median. Use constant time decision making to decide which is the true median relative to the rest of the values.

**Proof of Correctness:**

The values are always working towards a center point from the greater side of the lower median and the lower side of the greater median. As it converges towards the median, it splits its search index every iteration. Eventually it will yield 2 choices for which it can take constant time to decide which is the true median of total values.

**Recurrence Relation**:

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

T(n) = 2T(n/4) + 2

T(n) = 2(2T(n/4) + 2) + 2 = 4T(n/4) + 4 + 2

T(n) = 4 (2T(n/8) +2) + 4 + 2 = 8T(n/8) + 8 + 4 + 2

T(n) = 2^k T(n/2^k) + summation from i= 1 to k (2^i)

n/2^k = 1

n = 2^k

logn = k

T(n) = 2^logn T(n/2^logn) + 2logn = n\*T(1) + 2 logn = 2logn

Therefore, the runtime is theta(logn)

**Space Complexity**:

Would be 2 arrays to hold the data of the 2 lists.