Stephen Woodbury   4/20/17   CMPS 102

I have read and agree to the collaboration policy. Stephen Woodbury.
Grading: Homework Heavy
Collaborators: none

Assignment 1_4 : Divide and Conquer

Algorithm PseudoCode:

```
Median(n,a,b)
    if( n = 1 )
        return min( A(a),B(b) )
    k = ceiling(n/2)
    if( A(a+k) < B(b+k) )
        return Median(k, a + floor(n/2), b)
    else
        return Median(k, a, b + floor(n/2))
```

Proof of Correctness: Termination

Claim: *Algorithm terminates after at most log(n) recursive calls*

Pf: *Each time the function is recursively called, n is halved.*
There are only log(n) possible recursion calls before n=1
When n=1, the base case is fulfilled (n equaling 1).
Once the base case is fulfilled, the function returns.

Proof of Correctness: Median Accuracy

Claim: *Median Value in recursive call is the same as the median value in the original call*

Pf(via induction):

Let n' = original n

I) Base Case: range of elements included in Median(n,0,0)
ie, after zero recursive calls to Median
All elements are included, not are left out, therefor our median remains inside the range of elements.

II) Induction Step: Let the median value remain in the range of elements left after k recursive calls of Median(n,a,b)
This is our Induction Hypothesis.
[NTP: Median Value is in range of elements left over after k+1 recursive calls to Median(n,a,b) : Induction Conclusion]
Analyze k+1 calls to Median(n,a,b).
In order to have k+1 calls to Median, one must first have k recursive calls to Median. By our induction hypothesis, our median is in the range of elements left over from k-recursive calls to Median. One of two scenarios will occur:
Scenario 1)A(a+k) < B(b+k) is true, this calls *Median(k, a + floor(n/2), b) which by nature of the algorithm ensures the median is still within the range.*
*Scenario 2)*A(a+k) < B(b+k) is false, this calls *Median(k, a, b + floor(n/2)) which by nature of the algorithm ensures the median*

is still within the range.

Therefore at the last recursion call, our median is in our range of elements left over. Once the median is found, it is returned, all the way through the logn recursion calls called. This shows us that our median from our last recursion call is the same median from our original element range.

<u>Recurrence relation:</u>

T = T(ceiling(n/2)) + 3

1 operation for line 2

1 operation for line 4

1 operation for line 5

T(ceiling(n/2)) is the function recursively calling itself to solve itself.

T(ceiling(n/2)) + 3 = T(T(ceiling(n/4)) + 3) + 3

=T(T(T(ceiling(n/8)) + 3) + 3)) + 3....

This will yield 3(log(n).

Making Theta(log(n)) = Worst Case Run Time