Lecture 9: Efficiency Analysis Exercise Class

CS303: Algorithms

Last update: January 26, 2014

1 Review

- Efficiency Analysis: Asymptotic notations (O, Θ, Ω), Summation approach, substitution method, recurrence tree, and master theorem
- Algorithms: Fibonacci sequence generation, insertion sort, merge sort.

2 Asymptotic notations

1. Indicate whether the first function of each the following pairs has a smaller, same, or larger order of growth than the second function
   (a) $n(n+1), 2000n^2$
   (b) $log_2{}^2 n, logn^2$
   (c) $(n-1)!, n!$
   (d) $log_2 n, lnn$
2. Prove the following property
   $\Theta(\alpha g(n)) = \Theta(n)$ where α is a positive constant.
3. For each of the following functions, indicate the class using Θ notations
   (a) $(n^2+1)^{10}$
   (b) $\sqrt{10n^2+7n+3}$
   (c) $2^{n+1} + 3^{n-1}$

3 Efficiency Analysis

1. Find the efficiency of the following algorithm

```
Mystery(A[1,2,...,n])
value←A[1]
for i←1 to n
do
    if(A[i]>value)
        value←A[i]
done

return value
```

2. Find the efficiency of the following algorithm

```
Min(A[l,...r])
if(l==r) return A[l]
else
    temp1  ← Min(A[l,...⌊(l + r)/2⌋])
    temp2  ← Min(A[⌊(l + r)/2⌋ + 1,...,r])
    if(temp1≤ temp2) return temp1
    else return temp2
```

3. Use the recursion tree approach to find the efficiency of $T(n) = 9T(n/3) + n$ and prove your answer using the substitution method.

4. Use the master theorem to find the efficiency of the following algorithms

   - $T(n) = 3T(n/2) + n$
   - $T(n) = 3T(n/2) + nlgn$
   - $T(n) = 4T(n/4) + nlg^2n$
   - $T(n) = T(n/3) + 1$