CS 3343 Analysis of Algorithms – Spring 06

1/31/05

2. Homework

Due: Tuesday 2/14/05 before class. There is **no excuse date**.

1. **Recursive Mystery**

```
int mystery(int n){  // for n >= 0
  if(n==0) return 1;
  int m=n/3; //rounded down
  int tmp = mystery(m);
  tmp = tmp*tmp*tmp;

  // n-3*m is either 0,1,or 2
  for(int i=1; i<= n-3*m; i++){
    tmp = tmp*3;
  }
}
```

   (a) What does `mystery(n)` compute? Briefly justify your answer.

   (b) Clearly `mystery(n)` is a recursive algorithm. But is it also a divide-and-conquer algorithm? Briefly justify your answer.

   (c) Set up a runtime recurrence for `mystery(n)`.

   (d) What do you think this recurrence solves to? (No formal proof needed, but a brief convincing argument)

2. **Guessing and Induction**

   For each of the following recurrences find a good guess of what it could solve to. Make your guess as tight as possible. Use the recursion tree method to find your guess. Then prove that $T(n)$ is in big-Oh of your guess by induction (= substitution method; including base case and inductive case).

   Hint: You may want to use $\log_3 n$ instead of $\log_2 n$.

   Every recursion below is stated for $n \geq 2$, and the base case is $T(1) = 1$.

   (a) $T(n) = 3T(\frac{n}{3}) + 5$

   (b) $T(n) = 3T(\frac{n}{3}) + 7n$