Written Homework problems.

1. Suppose a recursive algorithm performs 2 recursive calls. Assume the first recursive call is of size at most 70% the original input size, and the second call is of size at most 25% of the original input size. In addition, the algorithm performs O(n) additional work after making these recursive calls. What is the big-Oh run time of this algorithm?

2. Suppose the above algorithm performs a recursive call of size 75% for the first recursive call, with everything else remaining the same. What is the big-Oh run time for this algorithm now?

3. Let F_n denote the nth Fibonacci number. Let A = $\begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}$. Show that A^n = $\begin{pmatrix} F_{n+1} & F_n \\ F_n & F_{n-1} \end{pmatrix}$. (Hint: use induction).

4. Use the equality proven in question 3 to design an algorithm to compute the nth Fibonacci number in O(log n) time.

5. Suppose Dr. WhyLie comes up to you and claims that he has invented a super-fast comparison based priority queue. The speed of the priority queue operations are as follows (n is the number of items currently in the priority queue):

   a. insert a new item in O(sqrt(log n)) time
   b. extract (remove and return) the smallest item from the priority queue in O(sqrt(log n)) time.

Explain why Dr. WhyLie must be lying.