CS 5200 PSET -1

Avah Banerjee

Due on: January 31, 11:59PM

Problem 1 Here is an alternate analysis of the Euclid's algorithm (computing $gcd(a, b)$). Suppose $s_n = a + b, s_{n-1} = b + (a \mod b), \ldots, s_1 = gcd(a, b)$ and take $s_0 = 0$. Then show that $a + b \geq F_n$, where F_n is the n^{th} Fibonacci number. Use the above observation to derive an upper bound on the number of recursive calls made by the Euclid's algorithm. Is this bound better than which was given in class?

Problem 2 Formally solve the recurrence relation: $T(n) = T(\sqrt[3]{n}) + O(\log \log n)$.

Problem 3 Let the matrix A_m be defined recursively as follows:

$$A_m = \begin{cases} [1] \text{ if } m = 0 \text{ (a } 1 \times 1 \text{ matrix)}, \\ \begin{bmatrix} A_{m-1} & A_{m-1} \\ A_{m-1} & -A_{m-1} \end{bmatrix} \text{ otherwise} \end{cases}$$

Take $n = 2^m$ and let B is a $n \times n$ matrix with integer entries. Assuming integer addition, multiplication etc. takes constant number of operations, devise a divide and conquer algorithm which uses $O(n^2 \log n)$ steps to compute the product $A_m B$.