Homework 2 - CSCI 181 - S20

For the following questions, when an algorithm is said to be $O(f(n))$, for simplicity assume that the time it takes to run this algorithm is $k.f(n)$ where k is a positive integer.

1. (20 pts)

   (a) We use a multiplication algorithm that takes time $O(\log^2(N))$ to multiply $N_1 \cdot N_2$ if $N_1 \approx N_2 \approx N$. If it takes 3 nanoseconds to multiply two 1000 bit numbers then how long would it take to multiply two 5000 bit numbers?

   (b) Let N be a large positive integer. The time it takes to factor N using trial division is $O(\sqrt{N})$. Assume that N' is a large positive integer and that the binary representation of N' has ten more bits than that of N. Assume that it takes 11 nanoseconds to factor N using trial division. Approximately how long would it take to factor N' using trial division?

2. (10 pts) Let's say we switch from 1024-bit RSA to 4096-bit RSA. How much longer does decryption take?

3. (20 pts)

   (a) The sum of integers from 1 to N is $N(N+1)/2$, that is $1+2+\ldots+N = N(N+1)/2$. Suppose a programmer does not know this formula and wants to compute the sum of integers from 1 to N. The programmer writes a program by adding 1 and 2 first and then getting the result and adding it by 3, and so on: $(((1+2)+3)+\ldots+N)$. Find the running time this will take in terms of N.

   (b) Find the running time in terms of N that it would take you to compute the sum using the formula $N(N+1)/2$. You'll see this is much faster.

4. (10 pts) Find the running time required to compute 6^N in terms of N. The computer program would compute $((((6 \cdot 6) \cdot 6) \cdots) \cdot 6)$.

5. (10 pts) Find the running time required to compute X^N, in terms of X and N. The computer program would compute $((((X \cdot X) \cdot X) \cdots) \cdot X)$.

6. (10 pts) Let F_n denote the nth Fibonacci number. We have $F_1 = F_2 = 1$ and for $i \geq 3$, $F_i = F_{i-1} + F_{i-2}$ (so $F_3 = 2, F_4 = 3, F_5 = 5, F_6 = 8, \ldots$). Recall $F_n \approx \alpha^n$, where $\alpha = (1+\sqrt{5})/2$. Find the running time of an algorithm that exactly finds the integer $\prod_{i=1}^{n} F_i$ using $(((F_1 \cdot F_2) \cdot F_3) \ldots \cdot F_n)$. Your answer should be $O()$ of a function of n and not have an F in it. Explain your answer. For simplicity, assume that you already have $F_1, \ldots, F_n$ in storage, so you don't have to worry about the time to compute them.