Problem 1
Give an $O(n \log n)$ integer multiplication algorithm, under suitable assumptions concerning the evaluation of sinusoid functions. **Hint:** Reduce integer multiplication to FFT for polynomial operations and be very clear about your assumptions.

Problem 2
Professor F. Lake tells her class that it is asymptotically faster to square an $n$-bit integer than to multiply two $n$-bit integers. Should you believe her?

Problem 3
Suppose that the adders within the butterfly operation of the FFT circuit sometimes fail in such a manner that they always produce a zero output, independent of their inputs. Suppose that exactly one adder has failed, but you do not know which one. Describe how you can identify the failed adder by supplying inputs to the overall FFT circuit and observing the outputs. How efficient is your method?

Problem 4
We have observed that the problem of evaluating a polynomial of degree $n-1$ at a single point can be solved in $O(n)$ time using Horner’s rule. We have also discovered that such a polynomial can be evaluated at all $n$ complex roots of unity in $O(n \log n)$ time using the FFT. We shall now show how to evaluate a polynomial of degree $n-1$ at $n$ arbitrary points in $O(n \log^2 n)$ time.
To do so, we shall use the fact that we can compute the polynomial remainder when one such polynomial is divided by another in $O(n \log n)$ time, a result that we assume without a proof. For example, the remainder of $3x^3 + x^2 - 3x + 1$ when divided by $x^2 + x + 2$ is $-7x + 5$: 
$$3x^3 + x^2 - 3x + 1 \mod (x^2 + x + 2) = -7x + 5$$

Given the coefficient representations of a polynomial $A(x) = \sum_{k=0}^{n-1} a_k x^k$ and $n$ points $x_0, x_1, \ldots, x_{n-1}$, we wish to compute the $n$ values $A(x_0), A(x_1), \ldots, A(x_{n-1})$. For $0 \leq i \leq j \leq n-1$, define the polynomials $P_{ij}(x) = \prod_{k=i}^{j} (x-x_k)$ and $Q_{ij}(x)$ as the remainder of $A(x)$ when divided by $P_{ij}(x)$, or $Q_{ij}(x) = A(x) \mod P_{ij}(x)$. Note that $Q_{ij}(x)$ has degree at most $j-i$.

a. Prove that $A(x) \mod (x-z) = A(z)$ for any point $z$.
b. Prove that $Q_{kk}(x) = A(x_k)$ and that $Q_{0,n-1}(x) = A(x)$.
c. Prove that for $i \leq k \leq j$, we have $Q_{ik}(x) = Q_{ij}(x) \mod P_{ik}(x)$ and $Q_{kj}(x) = Q_{ij}(x) \mod P_{kj}(x)$.
d. Give an $O(n \log n)$-time algorithm to evaluate $A(x_0), A(x_1), \ldots, A(x_{n-1})$.

Problem 5
Quicksort is a Las vegas randomized algorithm that runs in expected time $O(n \log n)$. Give a Monte Carlo randomized sorting algorithm that runs in time $O(kn \log n)$ and succeeds in sorting $n$ elements with probability at least $1 - 1/2^k$.