1. (15%) Given $a_0$, $a_1$, ..., $a_{n-1}$, the Fast Fourier Transform (FFT) algorithm can compute $A_0$, $A_1$, ..., $A_{n-1}$ in $O(n \log n)$ time complexity by the divide-and-conquer strategy, where

$$A_j = \sum_{k=0}^{n-1} a_k e^{i2\pi jk/n}, 0 \leq j \leq n - 1$$

$$= \sum_{k=0}^{n-1} a_k \omega^{kj}, \text{ where } \omega = e^{i2\pi/n}.$$  

Show the basic concept of the FFT algorithm.

2. (15%) Given a set of $n$ planar points $P_1, P_2, ..., P_n$, the Euclidean All Nearest Neighbor problem is to find the nearest neighbor of every $P_i$, $1 \leq i \leq n$. Show that the problem can be solved in linear time if the Voronoi diagram of the $n$ points is given.

3. (20%) We have the following definitions and theorem related to NP-completeness.

**Definition 1.** Let $A_1$ and $A_2$ be two problems. $A_1$ reduces to $A_2$ (written as $A_1 \preceq A_2$) if and only if $A_1$ can be solved in polynomial time, by using a polynomial time algorithm which solves $A_2$.

**Definition 2.** A problem $A$ is NP-complete if $A \in NP$ and every NP problem reduces to $A$.

**Cook’s Theorem.** NP=P if and only if the satisfiability (SAT) problem is a P problem.

Let SAT $\preceq B$, B $\preceq C$, C $\preceq D$, and D $\in NP$ for a given problem D. By the above definitions and theorem, show that D is NP-complete.

(Hint: You can first derive the reduction relationship of the SAT problem and every NP problem to complete the proof.)

4. Consider the problem to determine whether a given undirected graph with $n$ vertices contains a cycle of length $k$ as a subgraph. (a) (5%) Show that if $k$ is a part of the input instance, then this problem is NP-complete. (b) (15%) However, if $k$ is a constant, there is a naïve $O(n^k)$ time algorithm that solves this problem by checking all possible $k$ vertices of the graph. Design a more efficient algorithm for this problem.
5. Let $G$ be an undirected graph. Each edge $e$ in $G$ is assigned a probability $p_e$, $0 \leq p_e \leq 1$. The 
reliability of a path is the product of the probabilities of edges of the path. Design an efficient
algorithm to find a path with maximal possible reliability. (15%)

6. Given a positive integer $m$, and an array $p[1..n]$ of $n$ ($n \leq m$) positive numbers, find a way to
partition $m$ into one or more positive integers $j_1, j_2, \ldots, j_k$ (i.e. $j_1 + j_2 + \ldots + j_k = m$) such that the
sum $p[j_1] + p[j_1] + \ldots + p[j_k]$ is maximized over all possible partitions of $m$. (15%)