The following are some homework questions (more challenging than exam questions) from a pervious offering of CS 500. Sample solutions will be posted separately, so that you will be less tempted to peek.

Recommended background reading from *Introduction to Algorithms*, by Cormen, Leiserson, Rivest, Stein, Second Edition (CLRS):

NP-Completeness	pages 966–1021
Approximation Algorithms	pages 1022–1054

1. (from Exercise 8.20 in *Algorithms* by Dasgupta, Papadimitriou, and Vazirani) In an undirected graph $G = (V, E)$, we say $D \subseteq V$ is a *dominating set* if every $v \in V$ is either in D or adjacent to at least one vertex in D. In the DOMINATING SET problem, the input is a graph G and a budget b, and we must decide if G contains a dominating set of size at most b. Prove that this problem is NP-complete.

2. (from Problem 34-1 CLRS) **Independent Set**

   An **independent set** of a graph $G = (V, E)$ is a subset $V' \subseteq V$ of vertices such that each edge in E is incident on at most one vertex in V'. The **independent-set problem** is to find a maximum-size independent set in G.

   (a) Formulate a related decision problem for the independent-set problem, and prove that it is NP-complete. (Hint: Reduce from the clique problem.)

   (b) Suppose you are given a "black-box" subroutine to solve the decision problem you defined in part (a). Give an algorithm to find an independent set of maximum size. The running time of your algorithm should be polynomial in $|V|$ and $|E|$, where queries to the black box are counted as a single step.

   Although the independent-set decision problem is NP-complete, certain special cases are polynomial-time solvable.

   (c) Give an efficient algorithm to solve the independent-set problem when each vertex in G has degree 2. Analyze the running time, and prove that your algorithm works correctly.

   (d) Give an efficient algorithm to solve the independent-set problem when G is bipartite. Analyze the running time, and prove that your algorithm works correctly.

3. (From Exercise 8.14 in *Algorithms* by Dasgupta, Papadimitriou, and Vazirani) Prove that the following problem is NP-complete: given an undirected graph $G = (V, E)$ and an integer k, decide if G contains a clique of size k *as well as* an independent set of size k.

4. (from Exercise 35.2-3 CLRS) Consider the following **closest-point heuristic** for building an approximate traveling-salesman tour whose cost function satisfies the triangle inequality. Begin with a trivial cycle consisting of a single arbitrarily chosen vertex. At each step, identify the vertex u that is not on the cycle but whose distance to the cycle is minimum. (The distance of a vertex u to a cycle C is the minimum distance from u to a vertex in C.) Let v be the vertex on the cycle that is nearest to u. Extend the cycle to include u by inserting u just after v. Repeat until all vertices are on the cycle. Prove that this heuristic returns a tour whose total cost is not more than twice the cost of an optimal tour.