Problems to test basic knowledge of undergraduate algorithms.

Problem 1 We are given a set of n intervals on the real line, $I_1, I_2, \ldots, I_n$. Each interval I_j is specified as (ℓ_i, r_i) where ℓ_i is the left end point and r_i is the right end point. There is a positive weight w_j associated with I_j. Use dynamic programming to find the maximum weight subset of intervals that do not overlap. You can assume that the intervals are open intervals. Suppose we allow intervals to overlap but we require that the maximum number of intervals that can contain any point x on the real line is B for some constant $B > 1$. Can you generalize your algorithm to find a maximum weight subset of intervals that satisfy this relaxed condition? What is the running time of your algorithm?

Problem 2 Lipton and Tarjan showed that for any n vertex planar graph there is a balanced separator of size $O(\sqrt{n})$. A balanced separator is set of vertices whose removal partitions the graph into two disconnected graphs each with no more than $2n/3$ vertices. They also show that such a separator can be found in polynomial time. Use these facts to show how to compute a maximum independent set in G in $2^{O(\sqrt{n})}$ time. An independent set in a graph is a set of vertices that do not have edge between them.

Hint: Use divide and conquer and dynamic programming.

Problem 3 The classical $0, 1$ knapsack problem is the following. We are given a set of n items. Item i has two positive integers associated with it: a size s_i and a profit p_i. We are also given a knapsack of capacity B. The goal is to find a maximum profit subset of items that can fit into the knapsack. Use dynamic programming to obtain an exact algorithm for this problem that runs in $O(nB)$ time. Also obtain an algorithm with running time $O(nP)$ where $P = \sum_{i=1}^{n} p_i$. Note that both these algorithms are not polynomial time algorithms. Do you see why?

Now we consider a greedy algorithm for this problem. Assume without loss of generality that the items are numbered such that $s_1/p_1 \leq s_2/p_2 \leq \ldots \leq s_n/p_n$. The algorithm considers items in this order and places them in the knapsack until for the first time an item cannot be placed because it would violate the knapsack capacity. Let p_G be the profit of items packed by Greedy when it stops. Suppose for some $\epsilon < 1$, $s_j \leq \epsilon B$ for $1 \leq j \leq n$. Show that $p_G \geq (1 - \epsilon)O$ where O is the value of an optimum solution.