22C:31 Homework 1

Due in class on Tuesday, September 15th

1. Solve Problem 6, Chapter 1, pages 25-26, on ship scheduling for the *Peripatetic Shipping Lines, Inc.*. The easiest way to deal with this problem is this: model the problem as a version of the *stable matching* problem. Start by identifying ships as people of one gender and ports as people of the opposite gender. Then precisely identify preferences (defined over the set of ports) for ships and preferences (defined over the set of ships) for ports. Finally argue that a stable matching in this context immediately yields a set of truncations that respects the requirement that no two ships can be in the same port on the same day. Note that if you do all of this correctly, you don't have to find an algorithm. You can simply ask Peripatetic Shipping Lines, Inc. to use the Gale-Shapley algorithm for stable matching.

2. Consider the *weighted interval scheduling* problem described in Chapter 1, Page 14. There is a comment towards the end of the discussion on this problem that says "There appears to be no simple greedy rule that walks through the intervals, one at a time, making the correct decision in the presence of arbitrary values." "Check" this comment by providing simple counterexamples to the following simple greedy algorithms.

   (a) Repeatedly consider an interval with highest value, add it to the solution and delete it and all other intervals that overlap with it from the set of intervals being processed.

   (b) Define the *credit* of an interval as the ratio of its value to the number of other intervsls it overlaps. Intuitively, it seems like intervals with high credit should be picked up quickly to be in our solution because they provide a lot of value, yet do not rule out many other intervals. Here is the corresponding greedy algorithm. Repeatedly consider an interval with highest credit, add it to the solution and delete it and all other intervals that overlap with it from the set of intervals being processed.

   In both cases, assume that ties are broken arbitrarily.

3. Faced with a great deal of difficulty in designing correct algorithms for problems such as the *Maximum Independent Set (MIS)* problem (Chapter 1, Page 6), researchers have resorted to using approximation algorithms. Let c be a constant such that $0 < c < 1$. Then a *c-approximation algorithm* for the MIS problem is a polynomial-time algorithm that, for any input graph $G = (V, E)$, outputs a subset $S \subseteq V$ whose size is at least c times the size of a largest independent set. For example, if $c = 0.75$, then a c-approximation algorithm would be guaranteed to yield an independent set whose size is at least 75% of the size of the largest independent set.

   This problem guides you towards an example that shows that a simple and natural greedy algorithm for MIS is not a c-approximation for any constant c. Consider the greedy algorithm for MIS that repeatedly considers a vertex with smallest degree in the graph, adds it to the solution, and deletes it and all its neighbors from the graph.

   (a) Consider a graph with vertex set $\{a, b_1, b_2, b_3, c_1, c_2, c_3\}$. Connect vertex a to each b_i, $i = 1, 2, 3$, connect every b_i, $i = 1, 2, 3$ to every c_j, $j = 1, 2, 3$, and finally connect c_1, c_2, and c_3 to each other. For this graph show the independent set produced by the greedy algorithm and also the largest possible independent set.

   (b) Generalize this example, to come up with a graph for every positive integer $k \geq 3$ in which the degree-based greedy algorithm produces an independent set of size 2, whereas the graph contains an independent set of size k for any positive

(c) Argue why this family of graphs shows that the degree-based greedy algorithm is not a c-approximation for MIS for any constant c, $0 < c < 1$.

4. Problem 2, Chapter 2, Page 67.
5. Problem 4, Chapter 2, pages 67-68.
6. Problem 6, Chapter 2, pages 68-69.
7. Problem 8, Chapter 2, pages 69-70. This is a fun problem and the hint for part (a) is that $f(n) = \sqrt{n}$.
