Question 1

Suppose, totally hypothetically, that you’ve been captured by pirates. They are about to make you walk the plank, when, on a hunch, one of the scurvy scalawags asks you whether you know anything about approximation algorithms. Uh-oh, you think. It turns out they have something of an optimization problem and need your help. They would like to get a pirate fleet going, but they are having trouble deciding how many ships they need, and who should go on each ship. See, being pirates and all, certain pairs are prone to quarrelling and never seem to get anything done when they are together. On the other hand, some pirates work really well together, and it would be a shame to break up such teams.

The Captain gives you a list of those pairs of pirates who work well together, and a list of those who don’t. No pair appears on both lists, and some pairs might not appear on either. The problem is to partition the pirates into groups so that pirates who work well together are on the same ship, and pirates who do not work well together are on different ships. More formally, we will say that we have made a “good choice” for every pair of pirates who work well together and are assigned to the same boat, and for every pair who don’t work well that are placed on different boats. Similarly, a “bad choice” corresponds to splitting up pirates who work well, or putting together pirates who don’t.

(a) Show that there are instances in which it is impossible to assign pirates to boats without making at least one bad choice.

(b) The Captain, being something of a Renaissance man, has already proven that maximizing the number of good choices is NP-Complete. But he still needs a fleet, so he offers you your freedom if you come up with a 1/2 approximation to the problem. That is, you should come up with a deterministic (not randomized) algorithm to assign pirates to ships (you choose how many ships) that runs in polytime and always gets at least half of the best possible number of good choices. As a man of his word, even a silly solution will do, so long as you prove the 1/2 guarantee.

Question 2

Suppose you have a graph, and you need to color the nodes. Typically when you color a graph, you want to assign different colors to adjacent nodes. Any edge that has its two endpoints colored with different colors is said to be well colored, while any edge with the same color on both endpoints is poorly colored. But you only have 4 colors of paint, and you’re willing to tolerate a few poorly colored edges.

(a) Prove that for any constant integer $k$, it is NP-Complete to determine whether a graph $G$ can be 4-colored with at most $k$ poorly colored edges.
(b) Suppose the best 4 coloring of $G$ has $c^*$ well colored edges. Give a randomized algorithm to 4 color $G$ that has in expectation at least $\frac{3}{4}c^*$ well colored edges.

Question 3
In class we saw a simple greedy 2-approximation for the metric $k$-center problem. In this problem, we have $n$ cities, distances forming a metric between cities, and want to place $k$ centers at various cities so as to minimize the maximum distance of any city to its closest center. If the maximum distance, or covering radius, of an optimal solution is $r$, our algorithm is guaranteed to find a solution with covering radius at most $2r$. Prove that for any $\epsilon > 0$, a polytime approximation algorithm for this problem with approximation ratio $2 - \epsilon$ does not exist unless $P = NP$.

Hint: consider reducing from a problem we proved was hard on a previous problem set.

Bonus Question (Optional*)
*tricky, not worth much and no partial credit; basically just for fun if you’re bored done with the rest of the problems

The chromatic number of a graph $G$, denoted $\chi(G)$, is the minimum number of colors required to color the vertices such that no two adjacent vertices are assigned the same color. It is clear that if $G$ is a graph that contains a triangle (three nodes that are all connected to one another), then $\chi(G) \geq 3$. But suppose $G$ does not contain a triangle. We might still have $\chi(G) = 3$, as is the case if $G$ is a cycle with an odd number of nodes (at least 5). But are there triangle-free graphs with $\chi(G) > 3$? What is the largest value $\chi(G)$ can attain?