CS 577: Introduction to Algorithms	Homework 10
Out: 04/26/18	**Due: 05/03/18**

Ground Rules

- The homework is worth 5 points (out of a total of 100 points you can accumulate in this course).
- The homework is to be done and submitted in pairs. You can partner with someone from either section.
- The homework is due at the beginning of either lecture on the due date.
- No extensions to the due date will be given under any circumstances.
- Turn in your solution to each problem on a **separate sheet** of paper (or sheets stapled together), with your names clearly written on top.

Problems

1. **(Worth: 3 points. Page limit: 1 sheet; 2 sides)**

   A subset S of vertices in an undirected graph G is called triangle-free if, for every triple of vertices $u, v, w \in S$, at least one of the three edges (u, v), (u, w), (v, w) is absent from G.

   The problem *Large Triangle-Free Subset* (LTFS) takes as input a graph G and a number k, and outputs whether G has a triangle free subset of size $\geq k$.

   Show that LTFS is NP-hard.

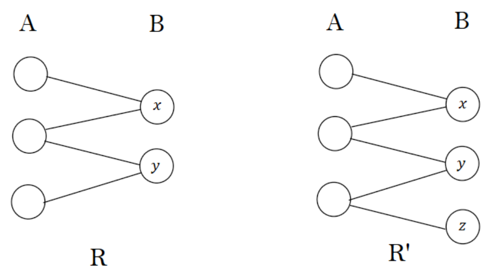
   (*Hint: Give a reduction from the Independent Set problem (IS) to LTFS.*)

(Turn over for problem 2)

2. **(Worth: 2 points. Page limit: 1 sheet; 2 sides)**

In the *Relation-to-Function* problem (RTF), you are given a relation R between two sets A and B, and you want to know whether you can erase some elements of B (and their corresponding tuples from R) so that R turns into a function on A.

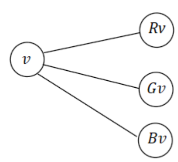
Pictorially, you are given a bipartite graph (with nodes $A \cup B$, and with edges between A and B), and you want to know whether there is a subset $C \subset B$ such that *every* node in A is a neighbor of *exactly one* node in C. For example, for the two graphs



the answer for R is "no" whereas for R' the answer is "yes" (since you can let $C = \{x, z\}$).

Show that RTF is NP-hard.

(*Hint: Give a reduction from the 3-Coloring problem to RTF. For every node, create the "node-gadget"*



which would capture the constraint that the node v is to be colored either Red, or Green, or Blue. Now think about how to create an "edge-gadget" that would capture the constraint that two ends of the edge must be different colors.)