CS577: Introduction to Algorithms (Spring 2018)	**Scribe:** Sam Vinitsky
HW10 Solutions	**Date:** 5/3/2018

Problem 1

To show that LFTS is NP-hard, we need to give a polynomial time reduction from some known NP-hard problem to LFTS. We will use Independent Set (IS) as mentioned in the hint. To reduce IS to LTFS, we construct a mapping from an instance (G, k) of IS to an instance of (G', k') of LFTS such that G has an independent set of size at least k if and only if G' has a triangle-free subset of size at least k'.

Intuitively, our reduction will substitute each edge $e = (u, v)$ in G by a triangle $\{u, w_e, v\}$ with a new vertex in G'. Given an instance (G, k) of IS where $G = (V, E), |V| = n, |E| = m$, create an instance (G', k') of LFTS as follows:

1. Let W be an empty set of vertices. Let F be an empty set of edges.
2. For each edge $e = (u, v) \in E$, add a vertex w_e to W.
3. For each edge $e = (u, v) \in E$, add edges (u, w_e) and (v, w_e) to F.
4. Let $G' = (V \cup W, E \cup F)$. (i.e. add all the vertices from W and edges from F)
5. Set $k' = k + m$.

Claim 1 *G has an independent set of size at least k $\iff$ G' has a triangle-free subset of size at least k'.*

Proof:

[$\Longrightarrow$] Suppose G has an independent set S of size at least k. Let $S' = S \cup W$. Since W contains one vertex for each of the m edges in G, we know $|W| = m$. Since S and W are disjoint, we know $|S'| = |S| + |W| \geq k + m$.

First, we will show S is triangle-free in G': Since S is an independent set in G, and in our construction of G' we didn't add any edges between vertices in G, then S must be an independent set in G' (by the definition of an independent set). Then in G' there are no edges in between any vertices in S. Then S must then also be triangle-free in G'.

Now, we will show that S' is triangle-free in G': Suppose for the sake of the contradiction that there is a triangle in S'. Then S', it must contain two vertices from S and one vertex from W (since by construction there are no edges between vertices in W, and S is triangle-free in G'). Therefore there are two vertices in S that have an edge between them in G'. This contradicts the fact that S is independent in G.

Therefore S' is indeed a triangle-free subset of G' of size at least $k' = k + m$.

[$\Longleftarrow$] Suppose G' has a triangle-free subset S' of size at least $k' = k + m$

If S' contains the whole set W, then $S = S' \setminus W$ must be an independent set of G'. If not, then there must be two adjacent vertices u and v in S, which means that $\{u, v, w_e\}$ is a triangle in S' (where $e = (u, v)$), which is a contradiction. Note that the size of S is $|S'| - |W| \geq k' - m = k$, so S is an independent set in G. of size at least k.

If S' does not contain the whole of W, there will always be another triangle-free subset S'' that contains all of W and has size at least k'. For each vertex w_e that is in W but *not* in S', do the following: (where $e = (u, v)$)

- If u and v are both not in S', add w_e to S'.
- If $u \in S'$, $v \notin S'$, add w_e to S'.
- If $u \notin S'$, $v \in S'$, add w_e to S'.
- If $u, v \in S$, remove u from S' and add w_e to S'.

For all of the possibilities above, we added w_e into S' without creating a triangle in S' and without decreasing the size of S'. We can add every $w_e \in W$ to S' to in this manner to create a new set S'' that is triangle-free and has size at least k'. Note that S'' now contains all of W, so we can use the same argument as before to show that $S = S'' \setminus W$ is an independent set in G of size at least k.

■

Problem 2

To show that RTF is NP-hard, we need to give a polynomial time reduction from some known NP-hard problem to LFTS. We will use 3-Coloring as mentioned in the hint. To reduce 3-Coloring to RTF, we construct a mapping from an instance G of 3-Coloring to an instance of (A, B, R) of RTF such that G is 3-colorable if and only if there exists a subset $C \subset B$ such every node in A is the neighbor of exactly one node in C.

Our reduction is as follows:

1. For each vertex v in G:
   (a) Add a "simple vertex" v to A
   (b) Add "color vertices" R_v, G_v and B_v to B.
   (c) Add edges from v to each of R_v, G_v and B_v.
2. For each edge $e = (u, v)$ in G:
   (a) Add "pair vertices" R_{uv}, G_{uv} and B_{uv} to A.
   (b) Add "auxiliary vertices" R'_{uv}, G'_{uv}, and B'_{uv}.
   (c) Add an edge from R_{uv} to R'_{uv}. Add edges from R_{uv} to each of R_u and R_v from step 1b. (Do the same for G_{uv} and B_{uv}.)

Claim 2 *$G = (V, E)$ is 3-colorable if and only if there exists a subset $C \subset B$ such every node in A is the neighbor of exactly one node in C.*

Proof:

[$\Longrightarrow$] Suppose $G = (V, E)$ is 3-colorable. Then there exists some 3-coloring such that no pair of adjacent vertices are assigned the same color.

We will now show (by construction) a subset $C \subset B$ such that every node in A is the neighbor of exactly one node in C:

1. For each vertex $v \in V$, v is colored a single color X in the 3-coloring. Add the color vertex X_v to C.
2. For each edge $e = (u, v)$ in G, u and v must be colored differently. Then there is some color X that does not color either of u or v. Add the auxiliary vertex X'_{uv} to C.

We will now prove that $C \subset B$ and every node in A is the neighbor of exactly one node in C:

Lemma 1 $C \subset B$

Proof:

By construction C contains only elements from B, and so $C \subseteq B$. For each vertex v in G, we added exactly one of R_v, G_v and B_v to C in step 1 of our construction (since v has exactly one color in the 3-coloring). Since all three of these vertices are in B, but only one is in C, we have that $C \neq B$, and thus $C \subset B$.

■

Lemma 2 *Every node in A is the neighbor of exactly one node in C*

Proof:

A consists of a simple vertex v for each vertex $v \in G$ and pair vertices R_{uv}, G_{uv} and B_{uv} for each edge $(u, v) \in G$.

Consider a simple vertex v in A. The neighbors of v in B are the color vertices R_v, G_v and B_v. In step 1 of our construction, we added exactly one of these to C (as above).

Consider a pair vertex R_{uv} in A. The neighbors of R_{uv} are the color vertices R_u, R_v and the auxiliary vetex R'_{uv}. Since u and v are adjacent, they must be different colors. We have the following cases:

- If both of the vertices are red, we would have a contradiction (since then our 3-coloring is invalid).
- If exactly one of the vertices (say, u) is red, then we would add R_u to C in step 1 of our construction. In this case, we wouldn't add the auxiliary vertex R'_{uv} in step 2, and so R_{uv}'s only neighbor in C is R_u.
- If neither of the vertices are red, we add the auxiliary vertex R'_{uv} in step 2, and so R_{uv} again has exactly one neighbor in C.

In all these cases, R_{uv} has exactly one neighbor in C. (The analysis is symmetric for B_{uv} and G_{uv}).

■

Therefore $C \subset B$ and every node in A is the neighbor of exactly one node in C, as desired.

[$\Longleftarrow$] Suppose that there exists a $C \subset B$ such that every node in A is the neighbor of exactly one node in C. We will prove that G has a valid 3-coloring.

We can explicitly construct the 3-coloring L as follows: Each simple vertex v in A has neighbors R_v, G_v and B_v. Exactly one of v's' neighbors is in C by assumption. Color $v \in V$ with the color corresponding to that neighbor.

Lemma 3 *L is a valid 3-coloring.*

Proof: By construction, each vertex v in V is colored exactly one color from $\{R, G, B\}$. This means that L is indeed a 3-coloring.

We will now show that no pair of adjacent vertices is colored the same color: Suppose for the sake of contradiction we have a pair of adjacent vertices u and v colored the same color (say, blue). Then B_u and B_v must both be in C. But B_{uv} is adjacent to both of these, which means it has two neighbors in C, which contradicts our original assumption that every vertex in A had exactly one neighbor in C. ■

■