EE360C: Algorithms
The University of Texas at Austin — Quiz #12
Dr. Christine Julien, Dr. Vallath Nandakumar — May 3, 2018 (*in class*)

Name: **EID:**

Quiz #12

Problem 1: NP or not NP?

You're currently a software engineer at SnapCat (A social media platform for cat lovers). You've been asked to lead the development of a completely new feature: synchronous photo broadcast, in which a single photo is simultaneously delivered to a pre-specified set of receivers.

For simplicity, assume a particular SnapCat user can be in only one of two states at any time: each user is either a *sender* or a *receiver*. At any time, there are n senders and m receivers. When a sender wants to perform a synchronous broadcast, he selects a subset of the receivers who should all simultaneously receive the photo. At the instant the photo is delivered, all of the designated receivers should be "assigned" to the given sender such that none of them are receiving any other photos and are instead all guaranteed to be focused on this single, shared experience.

Given that there may be many requests coming at the same time, and the sets of targeted receivers may be overlapping, we phrase the general ***Synchronous Sender Broadcast Scheduling*** **problem thusly:** Given sets of senders and receivers, the set of requested receivers for each sender, and a number k, is it possible to assign receivers to senders so that (1) each receiver is assigned to not more than one sender, (2) at least k senders will be active, and (3) every active sender is assigned all of its requested receivers.

For each of the following, either give a polynomial time algorithm or prove that the problem is NP-complete by starting from the *Independent Set* problem. If you give a polynomial time algorithm, you should also give the running time of your algorithm.

The Independent Set Problem. *Given a graph G and a number k, does G contain an independent set of at least k? In a graph $G = (V, E)$, a set of nodes $S \subseteq V$ is* independent *if no two nodes in S are joined by an edge.*

1. **[10 points]** The general *Synchronous Sender Broadcast Scheduling* problem described above.

Solution

This problem is NP-Complete.
The problem is in NP. Given a set of k senders, we can check in polynomial time that no receiver is assigned to more than one of them.
We next prove that *IndependentSet* $\leq_P$ *SynchronousSenderBroadcastScheduling*. Given an instance of the independent set problem, specified by a graph G and a number k, we create an equivalent sychronous sender broadcast scheduling problem. The receivers are edges, the senders are vertices, and the sender corresponding to vertex v wants to send the photo to all of the receivers that correspond to all of the edges incident on v.
Clearly, this reduction takes polynomial time to compute.
If there are k senders whose requested receivers are disjoint, then the k nodes corresponding to those senders form an independent set in the original formulation of the independent set problem. That is, any edge between these two nodes (which must not exist, since the nodes are part of an independent set) would correspond to a receiver they are both sending to.
If there is an independet set of size k, then the k senders corresponding to these nodes form a set of k senders that request to send to disjoint sets of receivers.

2. **[10 points]** The special case when $k = 2$, i.e., when we want to enable exactly 2 senders.

Solution

The case $k = 2$ can be solved by brute force: we just try all of the $O(n^2)$ possible pairs of senders and see if any pair has requested disjoint sets of receivers. This is a polynomial time algorithm.