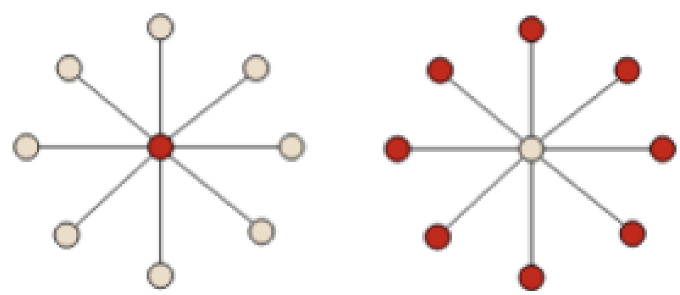
# 3500 hw 9

Casey Provitera

April 2024

## 1 Question 1(60 points)

An Independent Set problem:
Given a graph G = (V, E), we want to find subset of vertices in a graph, no two of which are adjacent. That is a set S of vertices such that for every two vertices in S, there is no edge connecting the two, and the $|S|$ is maximized.



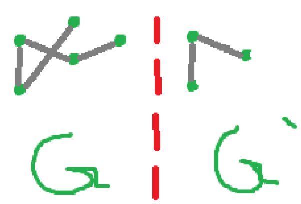
*Example:*

Two independent sets for the star graph (the right being maximum).
Prove Independent Set is NP-complete. Use VERTEX-COVER VC problem that proven to be NP-Complete and reduces it to the Independent Set problem. To remind you of VERTEX-COVER problem: Given a graph G = (V, E), we want to find subset of vertices C, such that all edges in G are covered by nodes in C, and the $|C|$ is minimized.

1. (30 Points) Prove that Independent Set is NP
   To find the answer to the independent set problem we need to try every possible set of vertices and check if they are valid under the independent set rules. Since we have to try every combination of vertices this means finding a solution can not be done with a polynomial time complexity, but rather some time complexity of factorial time because when you need to try all combinations of something that can be expressed as a factorial.
   But if we are given a solution to the problem we can in fact check in

polynomial time. This is because we know we only need to check each vertex a number of times equal to its degree.Which would give the time complexity of O(m*n) where n is the number of vertices and m is the degree of a vertex.
Since we can verify a solution in polynomial time but not find a solution in polynomial time this problem is classified as NP.

2. (30 points) Proof that Independent Set is NP-Hard (Use VC problem)
   Need to provide a way to transform the inputs from the vertex cover question into inputs for the independent set question.
   To prove that independent set is NP-Hard using the vertex cover problem we nee to be able to change vertex cover inputs so that if they are used as inputs in independent set we get the output of independent set to be what the out put should be of vertex cover.
   To do this we can take the input from the vertex cover problem and remove all vertices that have a degree of 1. After these vertices are removed we can input the new graph into the independent set problem. The output of the independent sent problem will now be what we know the output from vertex cover should be.

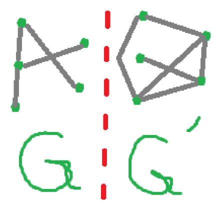
   We can see that this works from the example above. If G was the original input for vertex cover we convert to G' where we had removed two vertices. Now we use this G' as the input of independent set. Since independent set is maximizing it will choose the 2 vertices that do not share an edge. But those 2 vertices were exactly what we should get if we had originally input G into the vertex cover problem.
   Since we have now shown a way to convert the inputs for vertex cover to be inputs for independent set such that we get the answer we want for vertex cover from independent set we can say that independent set is at least as hard as vertex cover. Or that independent set is NP-Hard.

# 2 Question 2(40 points)

Let's proof of NP-completeness to the Independent Set problem again, but this time, let's utilize the Clique problem for this purpose. You don't need to reiterate the NP proof for the Independent Set problem; instead, begin by establishing that Independent Set is NP-hard through the polynomial reduction of the Clique problem to Independent Set.

1. (40 points) Proof that Independent Set is NP-Hard (Use Clique problem)

The Clique problem is solving for the exact opposite of the independent set problem.
Clique finds the maximum number of nodes that are all connected, while independent set finds the maximum number of nodes where none are connected.
This means if we want to convert the inputs for the vertex cover problem into inputs for the independent set problem where independent set will then give the outputs we want for vertex cover we can simply inverse the inputs.




For example imagine our clique input was the above graph G'. We then inverse it to get G as the input for independent set. With this input for the independent set problem we can get two possible out puts of the following sets of nodes. {Top Left, Bottom Left, Top Right} or {Bottom Right, Bottom Left, Top Right}. If we then translate each set of answers from independent set back onto G' which was the original input for clique we can see those vertices correlate to the two maximum clique on G'.
This is because when we take the inverse of a graph our new graph has edges connecting vertices that were not previously connected. When this happens Cliques are formed directly from the vertices that did not touch by edges before which is the condition we want for an independent set.
This can be used to see that the independent set problem is at least as hard as the Clique problem. Because the function we need to apply to the input of a Clique problem is an inverse functions where we create new edges in all places edges did not exist in the original and remove all original edges, where this can be done in polynomial time, we can classify independent set as a NP-Hard problem.