**Name: Taylor Heilman**

**CS 275: Spring 2016**

Say we're given a graph $G$nwith $n$ nodes and we want to find all possible graphs from those $n$ nodes.

First we need to find the maximum number of edges on that graph. To find the maximum amount of edges we use

$\binom{n}{2}$. we use this because you have $n$ number of vertices. That means, say there exists vertices $a, b$ that have a connecting edge. When counting all of the edges for vertex $a$ and all of the edges for vertex $b$, we count the edge connecting the 2 vertices twice. By using $\binom{n}{2}$ we correct the over counting of shared edges between two vertices.

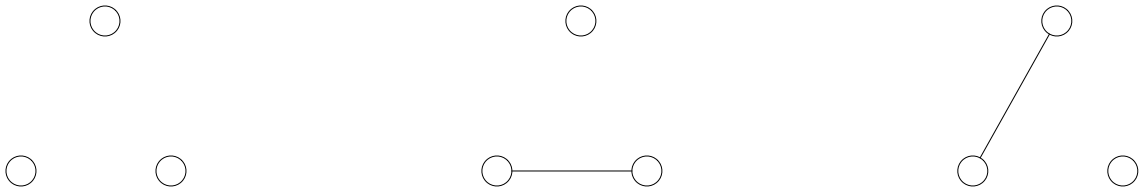
Now that we have the maximum number of edges we can find the maximum amount of graphs with $n$ vertices. according to Note 2.1 the total number of subgraphs in graph $G$ with $n$ vertices $= 2^k$. We use 2 because 2 vertices are need to create an edge. So by combing both out formulas we get $2^{\binom{n}{2}}$ = max graphs on $n$ vertices

Now if we want to find all graphs up to isomorphism we can break the problem down into smaller pieces. Say we want to do this for 3 vertices. We can break it down into 4 different cases, a case for each possible amount of total number of edges. With three vertices we have four cases, being 0,1,2,3. Next we find the total number of graphs for each case using the formula $2^{\binom{n}{2}}$. When doing this step we could create a dictionary to store each possible graph. Once we stored every possible graph we then must determine which of the graphs are isomorphic. If two graphs are isomorphic we will delete one and keep the other. Checking for isomorphism is the difficult part and is a process. Usually, to check if two graphs are isomorphic we check that the total number of edges and vertices in each graph is equal but since we know every graph in each case has the same number of edges and we know the total vertices is 3 we don't need to check either. First we check that each graph has the same degree sequence. If we confirm matching degree sequences we then check that any two vertices $u$ and $v$ of G are adjacent in G if and only if $f(u)$ and $f(v)$ are adjacent in a graph H. Next we check that corresponding vertices have the same degree. For instance if vertex $u$ has a degree of 3, than vertex $f(u)$ must also have that same degree. After that we check the degree of neighbors for a vertex of unique degree. Next we check the length

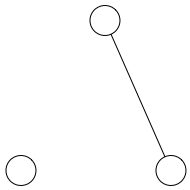
of the longest path in each graph and make sure the lengths match. Lastly we check the length of a geodesic between a pair of vertices with a given unique degree.

We check the things that take less computing steps first so if two graphs aren't isomorphic, we can prove it with an early test and no further testing is needed. To prove this algorithm works we can find all graphs on three nodes up to isomorphism. The maximum amount of edges is 3. This gives four cases. With 0 edges there is only one possible graph and that is 3 disconnected vertices. In case 1 we find 3 different graphs. In case 2 we also find 3 different graphs. Lastly, in case 3 we find 1 unique graph. Notice the symmetry in the maximum number of graphs possible. Below shows a table and diagrams of every possible graph up to isomorphism. To use this algorithm on a graph with 3 vertices took approximately 100-150 comparisons. This number will increase exponentially as the number of vertices increases.

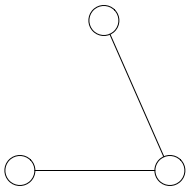
| Edges | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| Graphs | 1 | 3 | 3 | 1 |



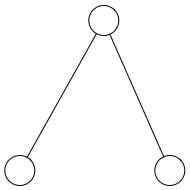
0.png 1.png 2.png



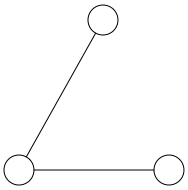
3.png



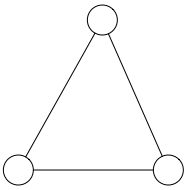
4.png



5.png



6.png



7.png