Exhaustive Search Minimal Graph Covering

The goal of this assignment is to write the **R** function `ESMGCA.R` that solves the minimal graph covering problem using an exhaustive search algorithm. The letters stand for "Exhaustive Search Minimal Graph Covering Algorithm."

As review, the minimal graph covering problem determines the smallest subset S of the set of vertices V of the graph (V, E) such that every vertex in V is adjacent to at least one vertex in S. For instance, for the graph in Figure 1, one minimal cover is the set $S = \{1, 4\}$

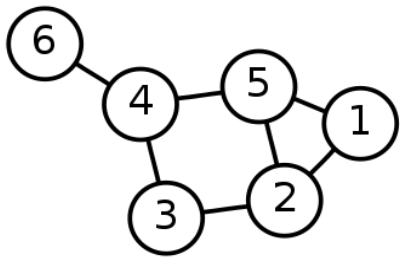

Figure 1: Graph with Six Vertices

because every vertex in the graph is adjacent to at least one vertex in S. Note that sets $\{2, 4\}$ and $\{4, 5\}$ are also minimal graph covers.

The final project uses the `edge2adj.R`, `Graph.Cover.R`, and `k.Subsets.R` functions that we have already written.

Input: *Edgelist.File*, a text string that contains the name of the file containing the list of edges of a graph.

Output: S, the vector containing the set of covering vertices.

Here's the idea:

1. Use the **R** function `edge2adj.R` to read the list of edges from a text file.

2. For each i in $1, 2, \ldots, |V|$, where $|V|$ represents the number of vertices in the graph, use the **R** function `k.Subsets.R` to generate all subsets of size i from a set of size $|V|$.

3. For each subset S of size i generated by `k.Subsets.R`, use the **R** function `Graph.Cover.R` to decide whether S covers the graph. If it does, return S. If it doesn't try the next set of size i.

Make sure that you submit just one file `ESMGCA.R` to Canvas. The other functions mentioned in this worksheet must be sub functions at the bottom of the file.

Here are several examples to try.

```
> Exhaustive.Search.Minimal.Graph.Covering.Algorithm("Figure01Graph.txt")
[1] 1 4
> Exhaustive.Search.Minimal.Graph.Covering.Algorithm("EdgeList01.txt")
[1] 1 2 9
> Exhaustive.Search.Minimal.Graph.Covering.Algorithm("EdgeList02.txt")
[1] 1 5 8
> Exhaustive.Search.Minimal.Graph.Covering.Algorithm("EdgeList03.txt")
[1] 1 6 8
> Exhaustive.Search.Minimal.Graph.Covering.Algorithm("EdgeList04.txt")
[1]  1  5  9 11 12 17
> Exhaustive.Search.Minimal.Graph.Covering.Algorithm("EdgeList05.txt")
[1] 5
```