Determining Whether a Given Set of Vertices Covers a Graph

The goal of this assignment is to write the **R** function `Graph.Cover.r` that determines whether a given set S of vertices covers the graph G.

Inputs:

- E, the edge list of a graph G
- S, a subset of G

Output: *TRUE/FALSE*, the function returns TRUE if S covers G, and FALSE if S does not cover G.

For instance, your function should be able to determine whether the set $S = \{3, 5\}$ covers the graph shown in Figure 1. There are several steps in this algorithm:

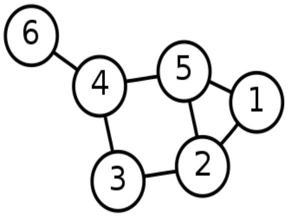

Figure 1: Graph with Six Vertices

1. Change the edge list into an adjacency list. For this example, your function should be able to change the edge list in Figure 2. The edge list in Figure 2 can be found in the file Figure01Graph.txt in Canvas. to the adjacency list in Figure 3.

1	2
1	5
2	3
2	5
3	4
4	5
4	6

Figure 2: Edge List Associated with Figure 1

1	2	5	
2	1	3	5
3	2	4	
4	3	5	6
5	1	2	4
6	4		

Figure 3: Adjacency List Associated with Figure 1

2. Initialize a Boolean vector *Cover.Index*, whose length equals the number of vertices in the graph, that represents whether a vertex has been covered. That is,

$$Cover.Index[i] = \begin{cases} \text{TRUE} & \text{if Vertex } i \text{ is adjacent to a vertex in } S \\ \text{FALSE} & \text{if Vertex } i \text{ is not adjacent to a vertex in } S \end{cases} .$$

   Initially, all elements of *Cover.Index* will equal FALSE.

3. Set all values of *Cover.Index* to TRUE that correspond to the vertices in S. **R** has a slick way to do this. Suppose that *Cover.Index* is a Boolean vector of length six, in which each element in *Cover.Index* equals FALSE. Also suppose that $S = \{3, 5\}$. Then, **R** can set the third and fifth element of *Cover.Index* to TRUE using the command

```
> Cover.Vertex[S] <- TRUE
```

   as in the example below.

```
> Cover.Vertex <- rep(FALSE,6)
> Cover.Vertex
[1] FALSE FALSE FALSE FALSE FALSE FALSE
> S <- c(3,5)
> Cover.Vertex[S] <- TRUE
> Cover.Vertex
[1] FALSE FALSE  TRUE FALSE  TRUE FALSE
```

4. Go through each component of the adjacency list and set the corresponding components of *Cover.Index* to TRUE using a “for” loop.

5. Finally, use **R**’s `sum` command to determine whether all the elements of *Cover.Index* are true. Here’s an example.

```
> Cover.Index <- c(TRUE,TRUE,TRUE,FALSE,TRUE)
> sum(Cover.Index)
[1] 4
> # If all the elements of Cover.Index are TRUE
> # then sum(Cover.Index) equals the length of Cover.Index
> sum(Cover.Index) == length(Cover.Index)
[1] FALSE
> Cover.Index <- c(TRUE,TRUE,TRUE,TRUE,TRUE)
> sum(Cover.Index) == length(Cover.Index)
[1] TRUE
```

Here are several examples.

```
> G01 <- read.table(file = "Figure01Graph.txt",sep = " ")
> Graph.Cover(G01,c(3,5))
[1] FALSE
> Graph.Cover(G01,c(2,4))
[1] TRUE
> E01 <- read.table("EdgeList01.txt",sep = " ",header = FALSE)
> Graph.Cover(E01,c(3,9))
[1] FALSE
> Graph.Cover(E01,c(1,2,9))
[1] TRUE
> E02 <- read.table(file = "EdgeList02.txt",sep = " ")
> Graph.Cover(E02,c(3,8,9))
[1] FALSE
> Graph.Cover(E02,c(1,5,8,9))
[1] TRUE
```