Akshay Gupta
Ankush Wadhwa
Kevin Tyler
CSCI 6444
Spring 2019

Project 1: Questions 6 and 7

* Please see find_longest_path(), contains_multiple_longest_paths(), build_neighborhood_matrix(), pretty_print_neighbors(), and determine_neighborhood_size() in the QuestionsSixAndSeven.R file.

** Note: for Questions 6.a, 6.c, 6.d, 6.e, and all of Question 7, we the same strategy to reduce the graph as we did for Questions 4 and Questions 5 (i.e.: *delete_vertices(roadGraph, which(degree(roadGraph) <= 8))*). However, Question 6.b deals with the longest path problem. According to Cormen et al in [1], the longest path problem is NP-complete.

Because the calculating the longest path is extremely computationally expensive, we therefore reduced the graph one step further for question 6.

```
reducedRoadGraph <- delete_vertices(roadGraph, which(degree(reducedRoadGraph) <= 12))
```

Part 6 (*"Determine the (a) central person(s) in the graph, (b) longest path, (c) largest clique, (d) ego, and (e) betweenness centrality and power centrality."*)

a. Is there more than one person with the most degrees? Yes.

```
> print(sprintf("Are there >1 nodes of maximum degree for road graph: %s",
+               length(which(degree(roadGraph) == max(degree(roadGraph)))) > 1))
[1] "Are there >1 nodes of maximum degree for road graph: TRUE"
```

b. Are there multiple longest paths? Yes.

```
> find_longest_path <- function(graph){
+     maxDistance <- 0
+     for(i in V(graph)){
+         for(j in V(graph)){
+             if(i == j){ next }
+             distance <- max(distances(graph, v = i, to = j))
+             if(distance > maxDistance){
+                 maxDistance <- distance
+             }
+         }
+     }
+     return(maxDistance)
+ }
>
> contains_multiple_lonest_paths <- function(graph, longestPath){
+     for( i in V(graph)){
+         for(j in V(graph)){
+             if(i ==j){ next }
+             if(max(distances(graph, v = i, to = j)) == longestPath){ return(TRUE)}
+         }
+     }
+     return(FALSE)
+ }
> #Find the longest path
> longestPath <- find_longest_path(reducedRoadGraph)
> print(sprintf("Longest path of reduced road graph: %s", longestPath))
[1] "Longest path of reduced road graph: Inf"
>
> #Are there multiple longest paths?
> print(sprintf("Are their mulitiple longest paths for reduced road graph %s",
+               contains_multiple_lonest_paths(reducedRoadGraph, longestPath)))
[1] "Are their mulitiple longest paths for reduced road graph TRUE"
```

c. Are there multiple cliques? Yes.

```
> #Are there multiple cliques?
> print(sprintf("Are their mulitiple cliques for subgraph %s",
+               clique_num(roadGraph) > 1))
[1] "Are their mulitiple cliques for subgraph TRUE"
```

d. Are there more than one person with the highest ego? Yes.

```
> #Find the ego of subgraph i
> egoSize <- ego_size(roadGraph)
> egoMaxIndex <- which.max(egoSize)
> egoNodeSize <- egoSize[egoMaxIndex]
> egoNodeLabel <- names(roadGraph[[egoMaxIndex]])
> multipleEgosOfMaximumSize <- (length(which(egoSize == egoNodeSize)) > 1)
> print(sprintf("Ego of road graph: Node %1s, which is of size %2s",
+               egoNodeLabel, egoNodeSize))
[1] "Ego of road graph: Node 6385, which is of size  5"
> print(sprintf("Are there multiple node with the highest ego? %s", multipleEgosOfMaximumSize))
[1] "Are there multiple node with the highest ego? TRUE"
```

e. What is the difference in betweenness centrality vs. power centrality for the cases you find? Consider comparing the nodes that are members of each set. Are there common nodes?

Betweenness centrality is number of shortest paths that flow through a given node, while power centrality refers to the number of directed edges that point to a given node (calculated recursively).

While the two nodes of greatest centrality are different, there are common nodes in each set.

```
> #Find betweenness centrality
> print(sprintf("Betweeness centrality of road graph: %s",
+               names(which.max(betweenness(roadGraph)))))
[1] "Betweeness centrality of road graph: 6385"
>
> #Find the power centrality
> print(sprintf("Power centrality of road graph: %s",
+               names(which.max(power_centrality(roadGraph, exponent = .7)))))
[1] "Power centrality of road graph: 5771"
```

Part 7 (*"Find the 20 nodes with the largest networks, e.g., having the greatest diameters. Do any of these circles overlap?"*):

	Level One Neighborhood	Level Two Neighborhood	Level Three Neighborhood
161	160, 162, 164, 165	161, 163, 172, 165, 169, 164	160, 162, 164, 165, 171, 161, 170, 169
162	161, 172	160, 162, 164, 165, 171	161, 163, 172, 165, 169, 164
164	161, 165, 169	160, 162, 164, 165, 161, 170	161, 163, 172, 165, 169, 164, 160, 162
165	161, 164	160, 162, 164, 165, 161, 169	161, 163, 172, 165, 169, 164, 160, 162, 170
3356	3355, 3357	3353, 3356, 3371	3355, 3357, 3367
3712	3711, 3713	3709, 3712, 3718	3711, 3713, 3716, 3719
3357	3356, 3371	3355, 3357, 3367	3353, 3356, 3371
3713	3712, 3718	3711, 3713, 3716, 3719	3709, 3712, 3718, 3719, 3716
4354	4353, 4361	4354, 4355, 4360	4353, 4361, 4359, 4362
4361	4354, 4360	4353, 4361, 4359, 4362	4354, 4355, 4360
10672	10665, 10673	10669, 10672, 10674	10665, 10673, 10671
10673	10672, 10674	10665, 10673, 10671	10669, 10672, 10674
4188	3905, 4186, 4187, 4705	4188, 5008	3905, 4186, 4187, 4705, 5009
3624	3360:3361	3624, 3625, 3364	3360:3361
3718	3713, 3716, 3719	3712, 3718, 3719, 3716	3711, 3713, 3716, 3719, 3718
3190	3189, 3333	3190, 3331	3189, 3333, 3332
4360	4359, 4361, 4362	4360, 4354	4359, 4361, 4362, 4353
169	164, 170	161, 165, 169	160, 162, 164, 165, 161, 170
3371	3357, 3367	3356, 3371	3355, 3357, 3367
3331	3332:3333	3331, 3190	3332, 3333, 3189

Yes, some of the circles overlap. The commons nodes are:

- **Node 6384 is shared by nodes 6, 62, 64, 65 and 69**
- **Node 6385 is shared by nodes 6, 6385, 62, 64, 65 and 69**
- **Node 6487 is shared by nodes 6, 62, 64, 65 and 69**
- **Node 6452 is shared by nodes 6, 62, 64 and 65**
- **Node 6460 is shared by nodes 6 62, 64, 65 and 69**
- **Node 6483 is shared by nodes 6, 62, 64, 65, and 69**
- **Node 6478 is shared by nodes 6, 62, 64, 65, and 69**
- **Node 6480 is shared by nodes 6, 64, 65, and 69**
- **Node 6484 is shared by nodes 6 and 62**
- **Node 6488 is shared by nodes 6, 62, 64 and 65**
- **Node 505898 is shared by nodes 3356, 3357 and 3356**
- **Node 505897 is shared by nodes 3357 and 337**
- **Node 505902 is shared by nodes 3356, 3357, and 337**
- **Node 506206 is shared by nodes 3356, 3357, and 337**
- **Node 534837 is shared by nodes 3356, 3357 and 337**
- **Node 534232 is shared by nodes 3356, 3357 and 337**

- **Node 542633 is shared by nodes 372", 542633 and 373**
- **Node 54256 is shared by nodes 372, 373 and 378**
- **Node 54257 is shared by nodes 372, 373 and 378**
- **Node 542632 is shared by nodes 372, 373, 378**
- **Node 542737 is shared by nodes 372, 373 and 378**
- **Node 542653 is shared by nodes 372, 373 and 378**
- **Node 58785 is shared by nodes 4354, 436, 4360**
- **Node 587959 is shared by nodes 4354, 436, and 4360**
- **Node 587957 is shared by nodes 4354, 587957 and 436**
- **Node 587946 is shared by nodes 4354, 436 and 4360**
- **Node 587947 is shared by nodes 4354, 436 and 4360**
- **Node 587958 is shared by nodes 4354", 436 and 4360**
- **Node 587963 is shared by nodes 4354, 436 and 4360**
- **Node 474975 is shared by nodes 0672 and 0673**
- **Node 474889 is shared by 0672 and 0673**
- **Node 474988 is shared by 0672 and 0673**
- **Node 474977 is shared by 0672 and 0673**
- **Node 47498 is shared by 0672 and 0673**
- **Node 474984 is shared by 0672 and 0673**
- **Node 49648 is shared by 390 and 333**
- **Node 50485 is shared by 390 and 333**

References

[1] Cormen, Thomas H.; Leiserson, Charles E.; Rivest, Ronald L.; Stein, Clifford (2001), Introduction To Algorithms (2nd ed.), MIT Press, p. 978, ISBN 9780262032933.