

# Dijkstra's Algorithm

Knox Game Design

January 2022

Levi D. Smith

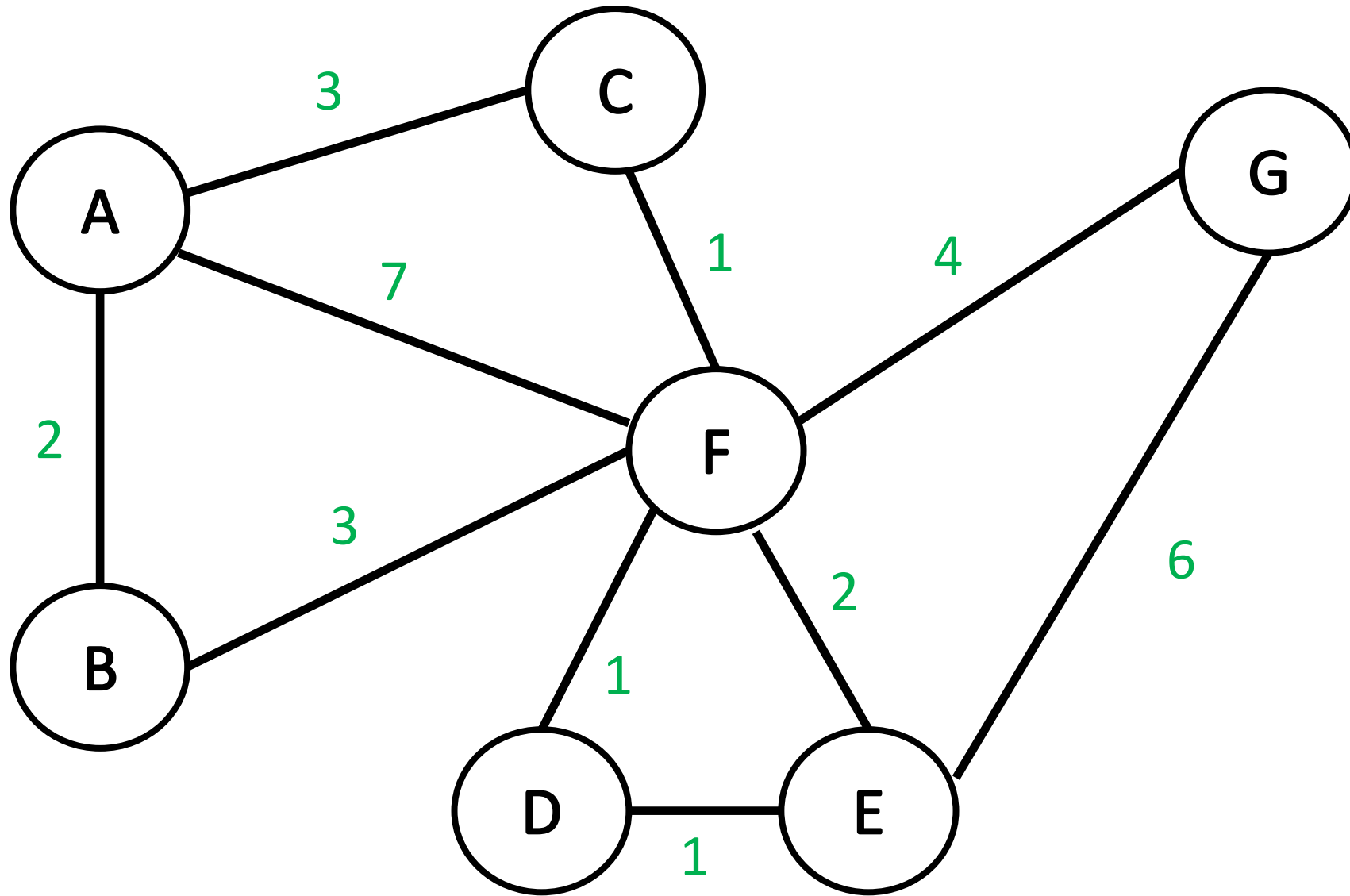
# What is Dijkstra's Algorithm

- Weighted graph
  - Nodes
  - Edges (connect nodes)
- Undirected
- Find the Shortest Path from starting point to all nodes
  - Not necessarily MST (minimum spanning tree) - Use Prim's instead

# Why is it useful

- RTS (real time strategy) path finding
- Sandbox game, pointer to objective
- Roads between intersections
- Tunnels between train stations
- Board game, shortest path to goal

Example



\*Not to scale

Start - A

Shortest path tree starts empty

Distance is all infinite for each node

Shortest Path: ()                      Distance: (INF, INF, INF, INF, INF, INF, INF)

Starting node distance to zero

Shortest Path: ()                      Distance: (0, INF, INF, INF, INF, INF, INF)

While Shortest Path does not include all nodes

Pick node with minimum from Distance that is not in Selected and add it to Selected

Update Distances with values from start to adjacent nodes from picked node.

Update if the total distance from start is less than current distance

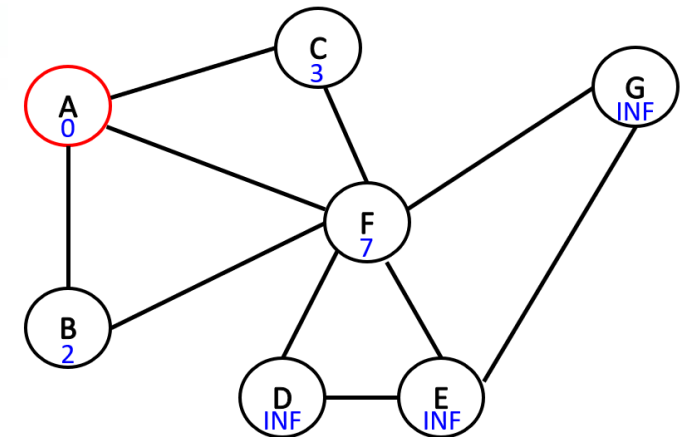
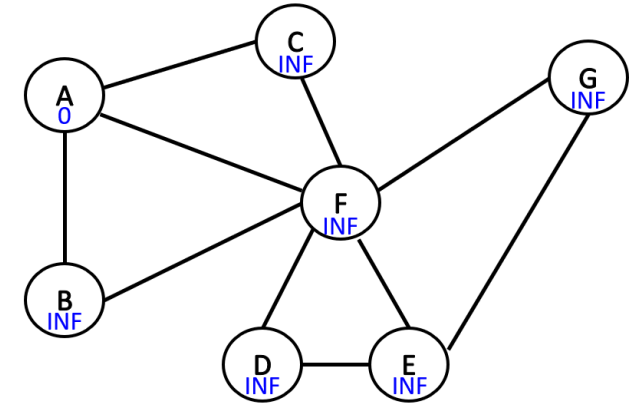
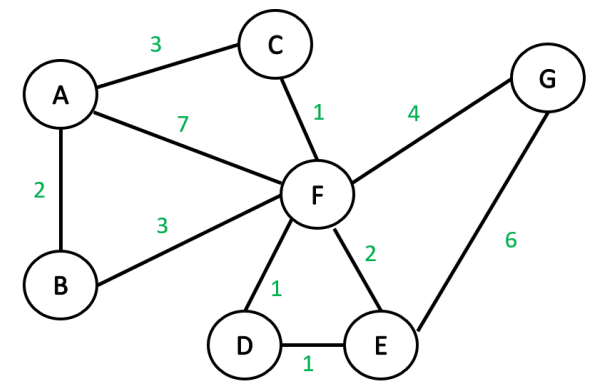
Updating Distance for already selected nodes is not important

Pick A (distance = 0)

Selected: (A)

Adjacent B, C, F      Update B, C, F

Distance: (X, 2, 3, INF, INF, 7, INF)



# KNOWLEDGE

# GOALS

# PROBLEMS

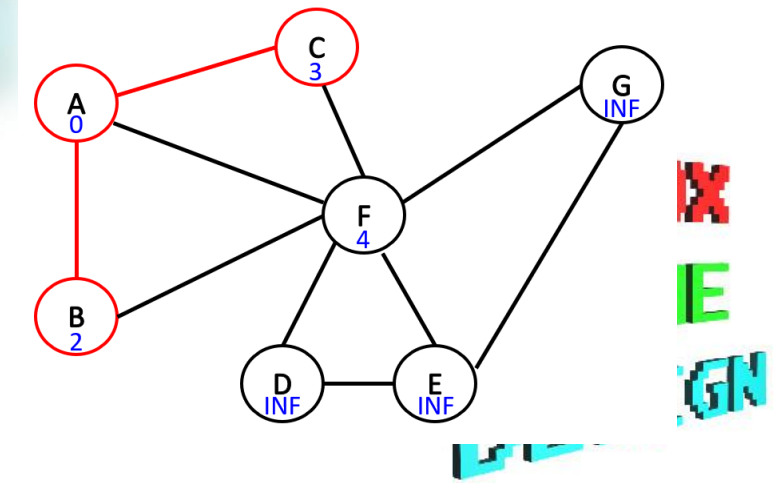
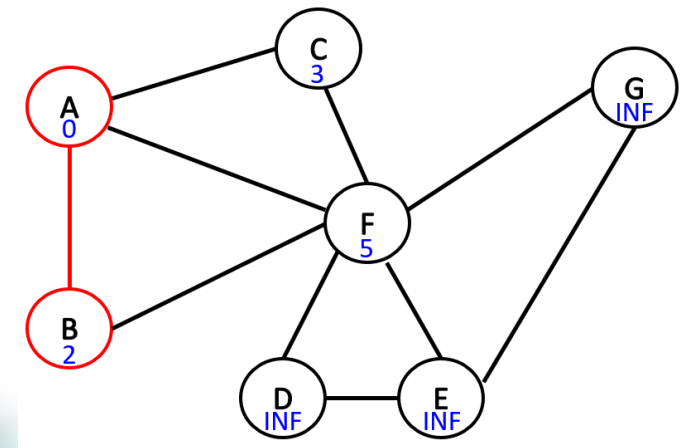
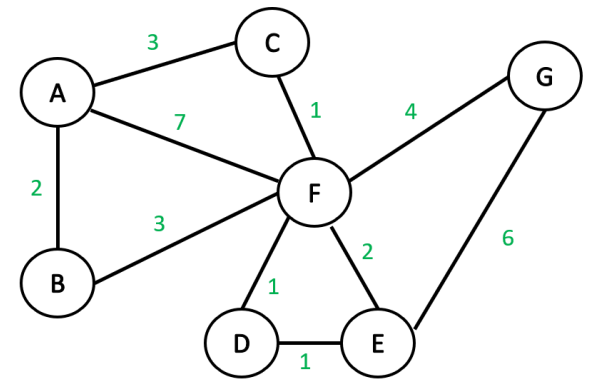
# STRATEGIES

Pick B (distance = 2)  
Selected: (A, B)  
Path: (A-B)

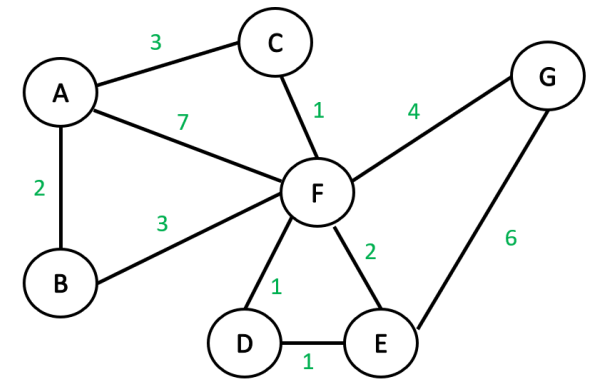
Adjacent F      Update F (5 < 7)  
Distance: (X, X, 3, INF, INF, 5, INF)

Pick C (distance = 3)  
Selected: (A, B, C)  
Path: (A-B, A-C)

Adjacent F      Update F (4 < 5)  
Distance: (X, X, X, INF, INF, 4, INF)

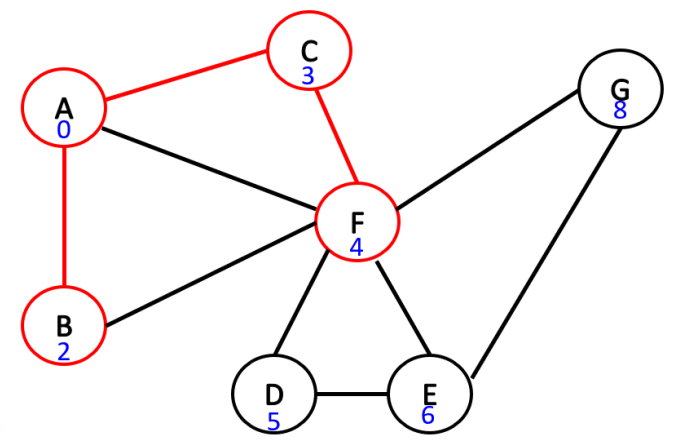


# KNOWLEDGE



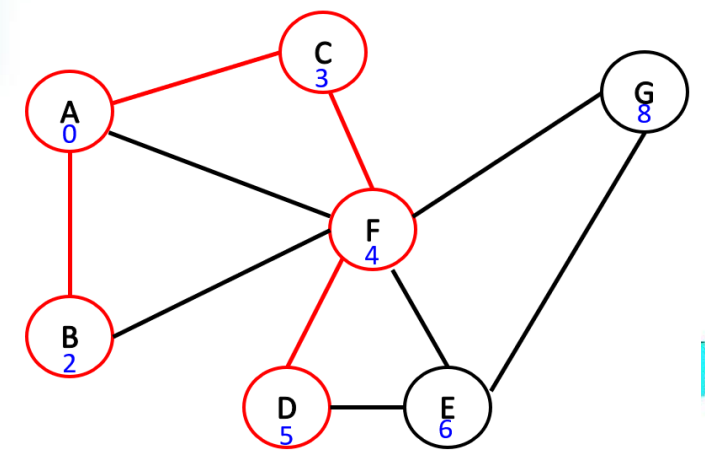
Pick F (distance = 4)  
Selected: (A, B, C, F)  
Path: (A-B, A-C-F)

Adjacent D, E, G    Update D, E, G  
Distance: (X, X, X, 5, 6, X, 8)

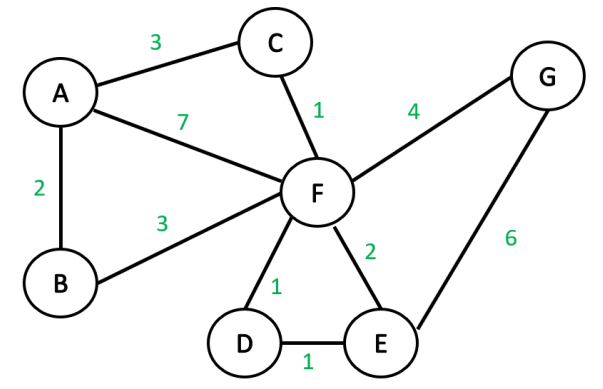


Pick D (distance = 4)  
Selected: (A, B, C, D, F)  
Path: (A-B, A-C-F-D)

Adjacent E    Don't update E (6 = 6)  
Distance: (X, X, X, X, 6, X, 8)

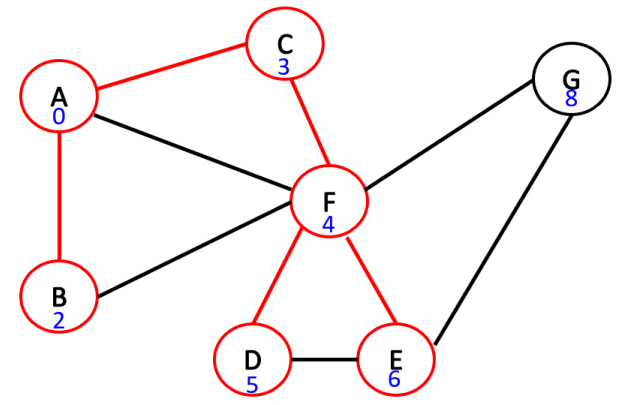


# KNOWLEDGE



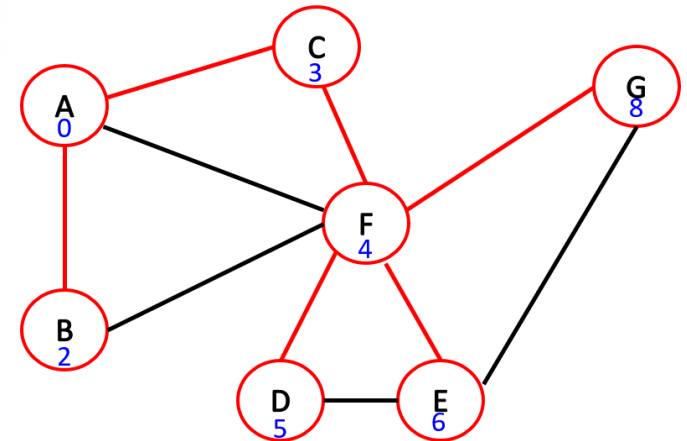
Pick E (distance = 6)  
Selected: (A, B, C, D, E, F)  
Path: (A-B, A-C-F-D, A-C-F-E)

Adjacent G  
Don't update G (12 > 8)  
Distance: (X, X, X, X, X, X, 8)



Pick G (distance = 8)  
Selected: (A, B, C, D, E, F, G)  
Path: (A-B, A-C-F-D, A-C-F-E, A-C-F-G)

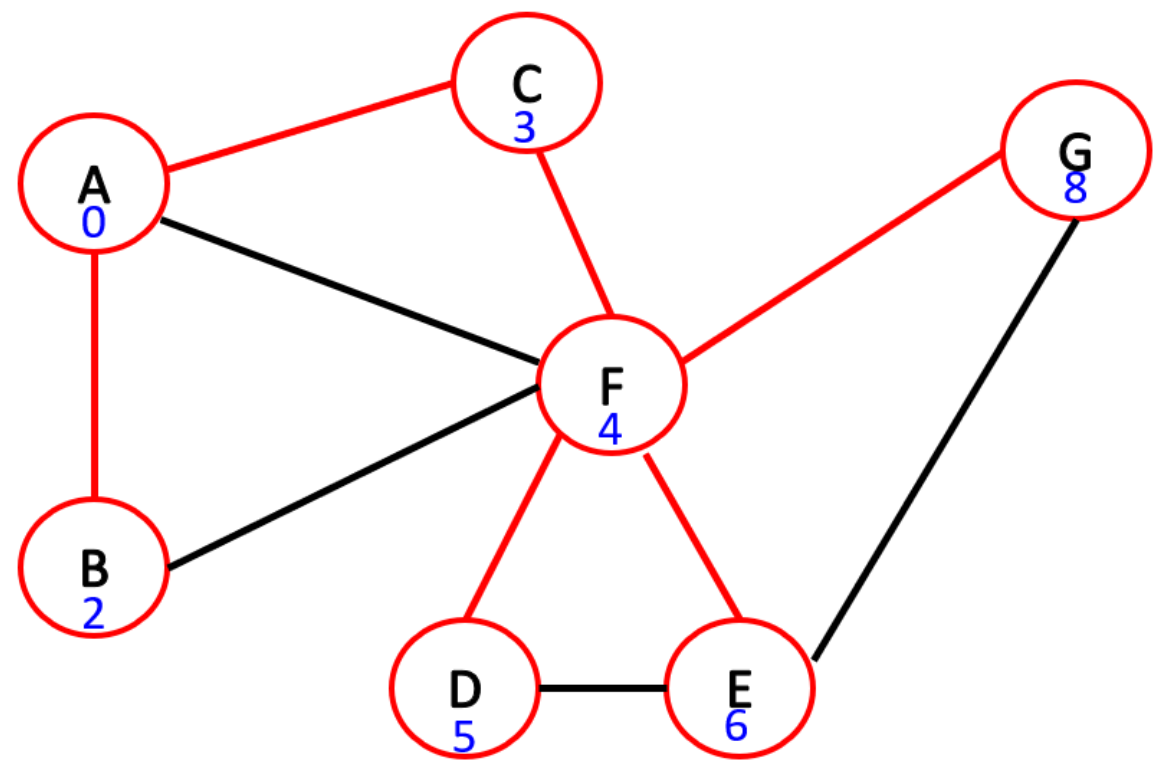
Adjacent None



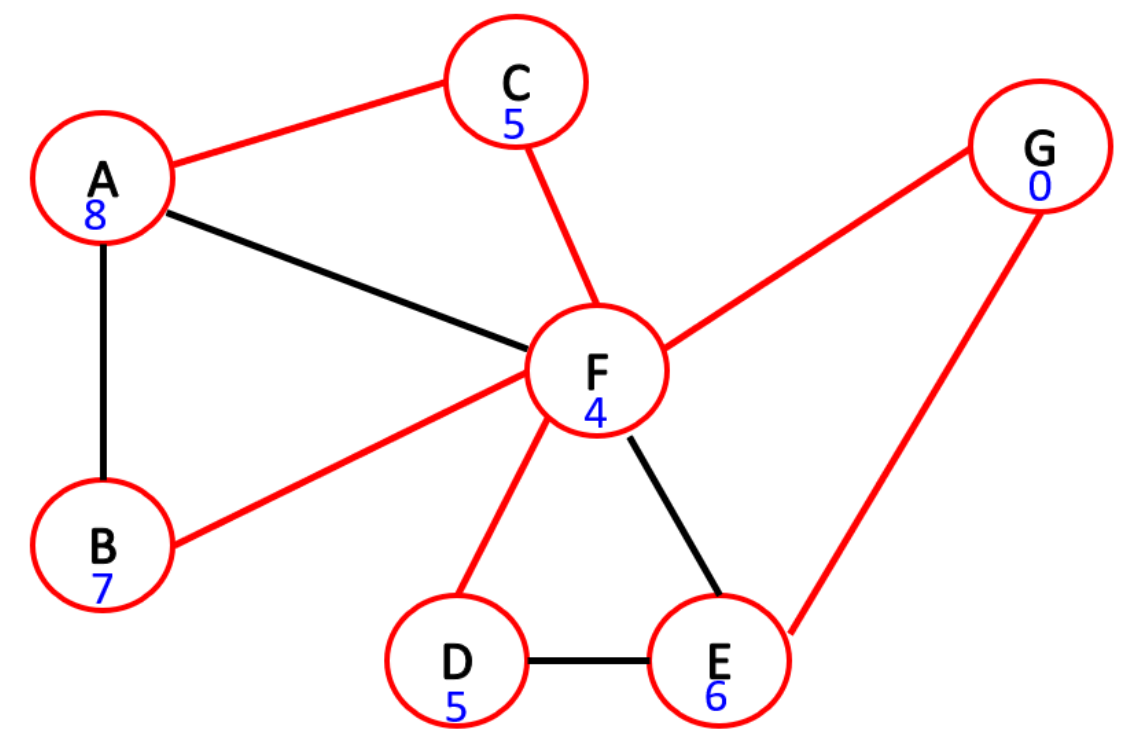


# CONTEXT

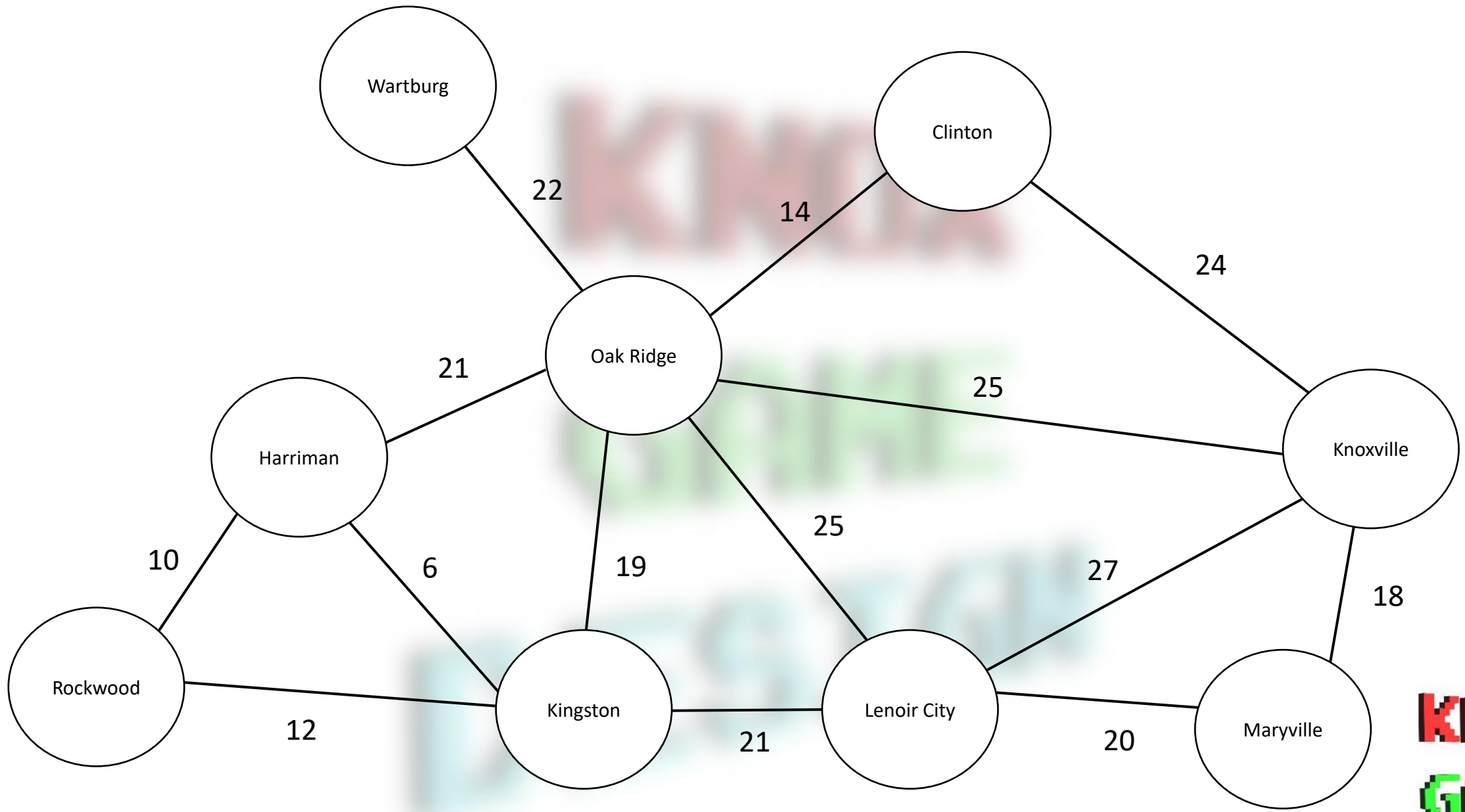
Starting Node A



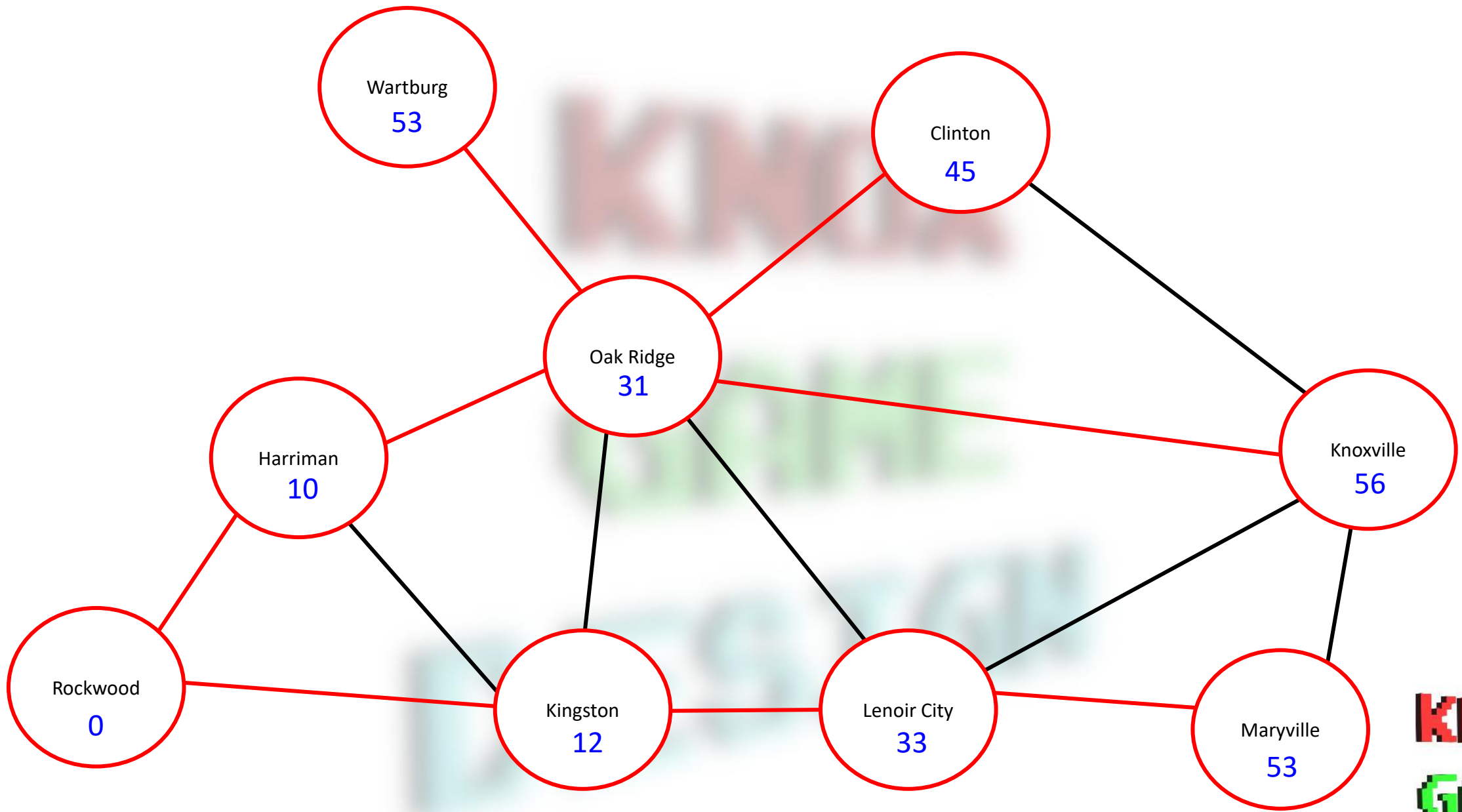
Starting Node G



GAME  
DESIGN

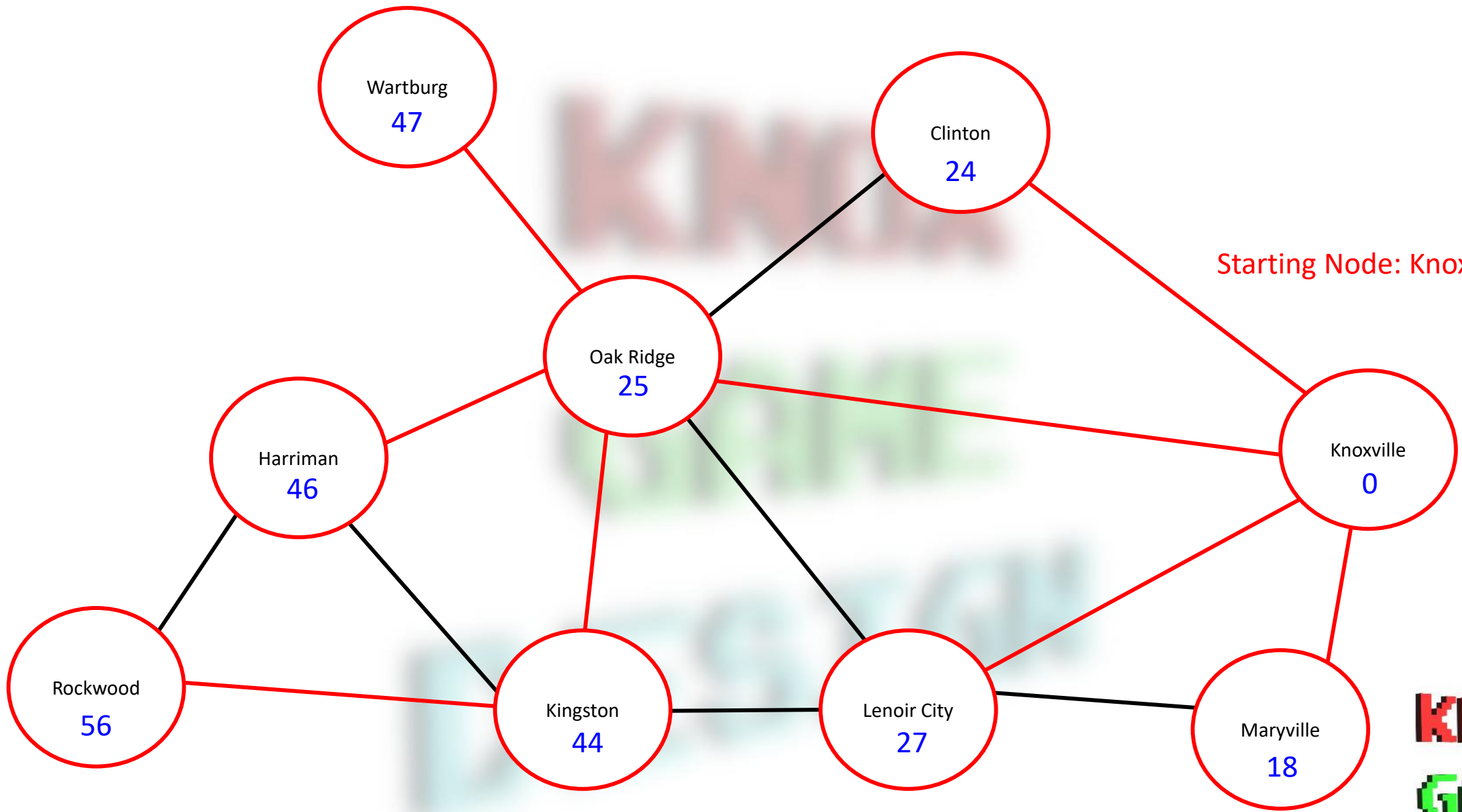


**KNOX**  
**GAME**  
**DESIGN**



Starting Node: Rockwood

**KNOX**  
**GAME**  
**DESIGN**



Starting Node: Knoxville

**KNOX**  
**GAME**  
**DESIGN**