



NETWORK ANALYSIS IN PYTHON I

Degree centrality



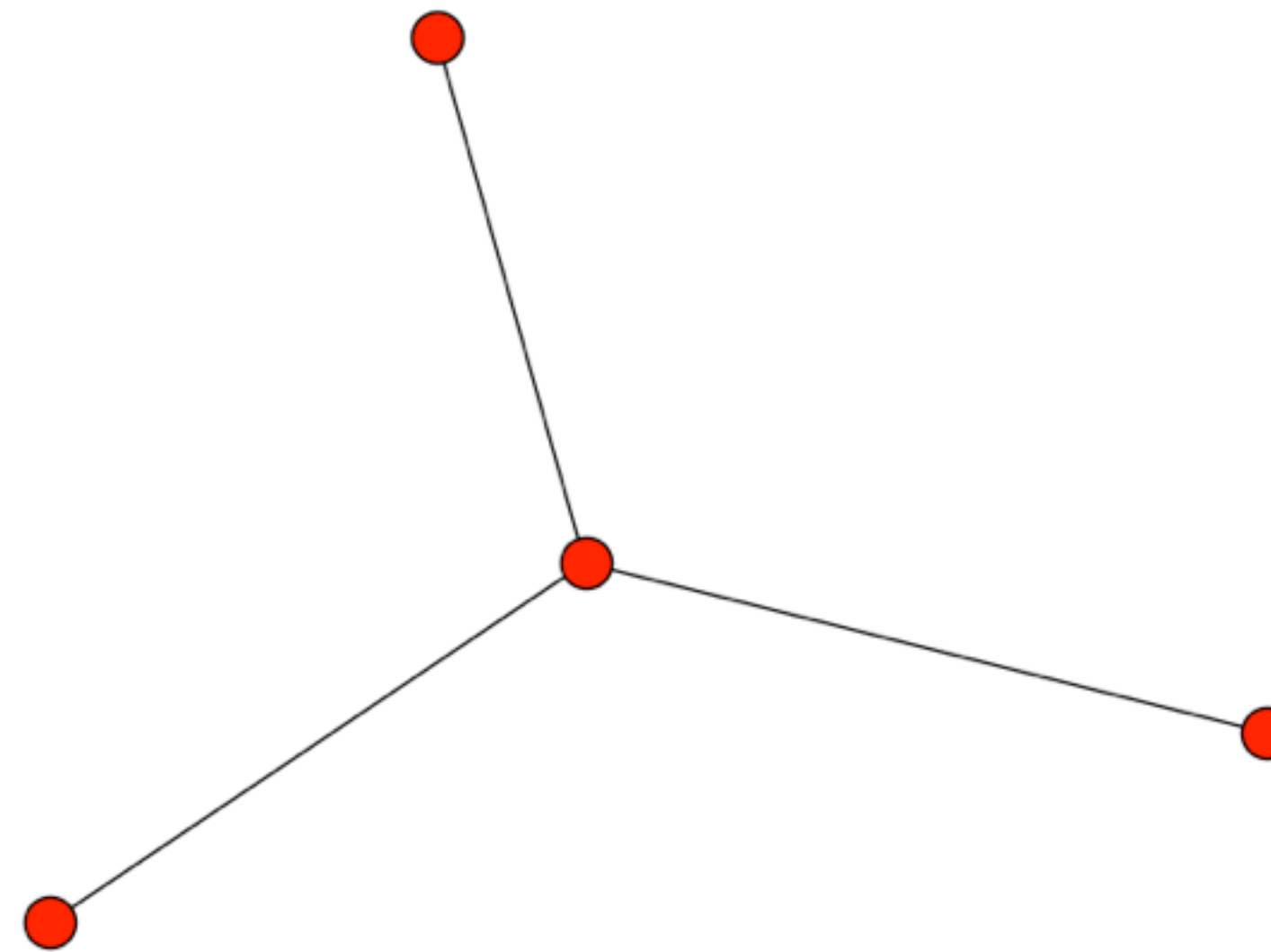
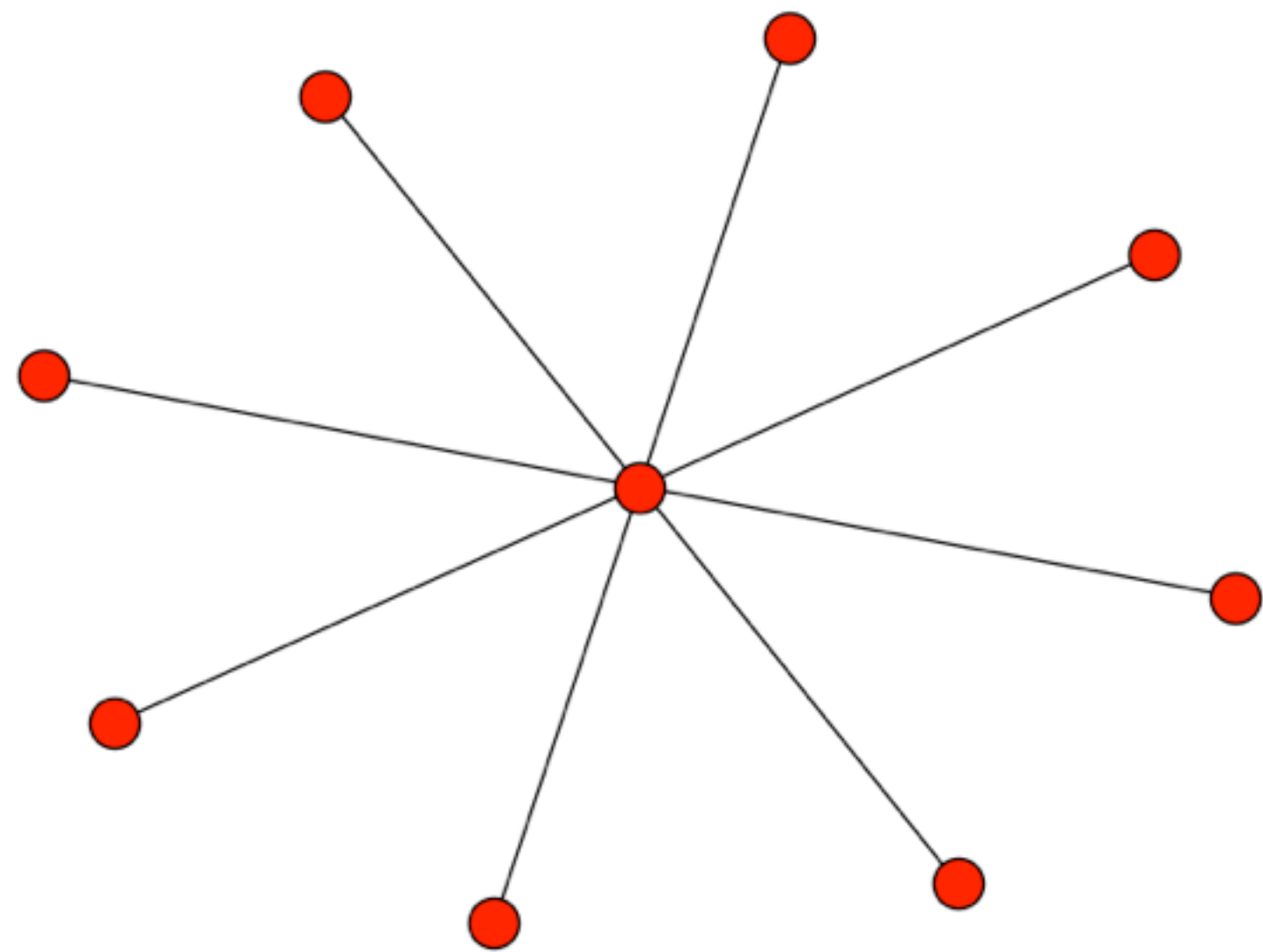
Important nodes

- Which nodes are important?
 - Degree centrality
 - Betweenness centrality



Important nodes

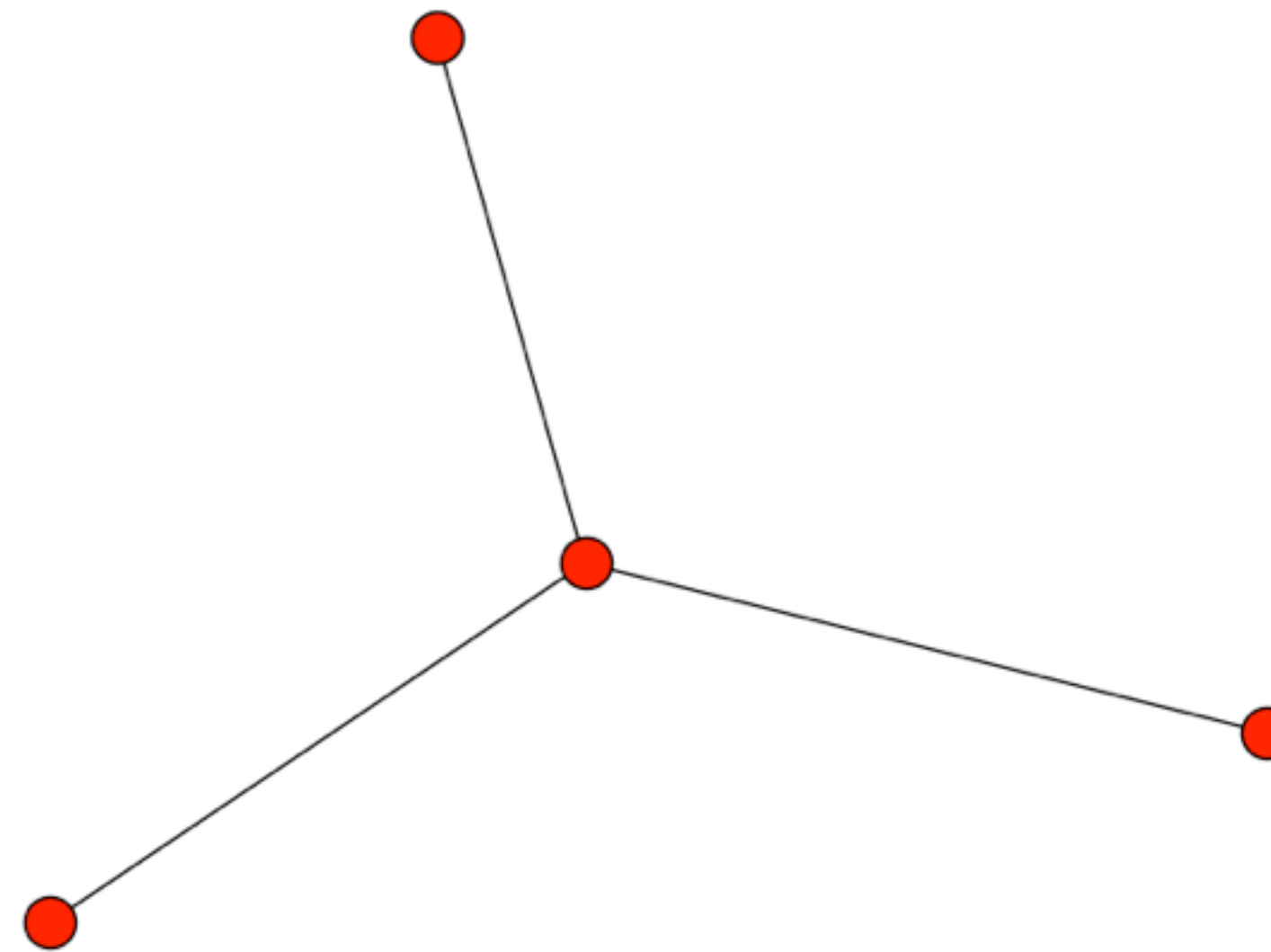
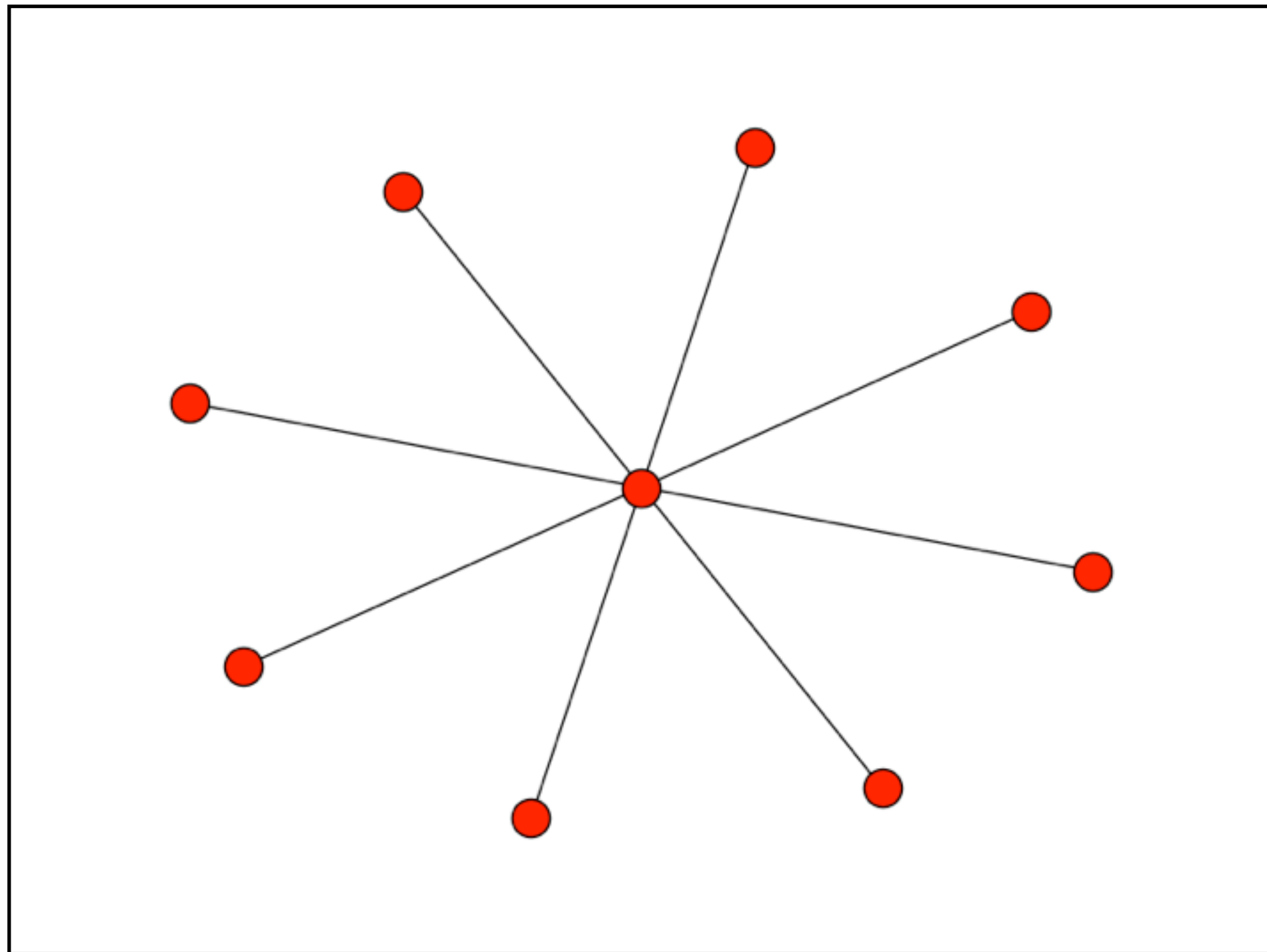
- Which centre node might be more important?





Important nodes

- Which centre node might be more important?





Degree centrality

- Definition:

Number of Neighbours I Have

Number of Neighbours I Could Possibly Have

- Examples of nodes with high degree centrality:
 - Twitter broadcasters
 - Airport transportation hubs
 - Disease super-spreaders



Number of neighbors

```
In [1]: G.edges()
Out[1]: [(1, 2), (1, 3), (1, 4), (1, 5), (1, 6), (1, 7), (1, 8),
(1, 9)]

In [2]: G.neighbors(1)
Out[2]: [2, 3, 4, 5, 6, 7, 8, 9]

In [3]: G.neighbors(8)
Out[3]: [1]

In [4]: G.neighbors(10)
.....
NetworkXError: The node 10 is not in the graph.
```



Degree centrality

```
In [5]: nx.degree_centrality(G)
```

```
Out[5]:
```

```
{1: 1.0,  
 2: 0.125,  
 3: 0.125,  
 4: 0.125,  
 5: 0.125,  
 6: 0.125,  
 7: 0.125,  
 8: 0.125,  
 9: 0.125}
```



NETWORK ANALYSIS IN PYTHON I

Let's practice!



NETWORK ANALYSIS IN PYTHON I

Graph algorithms



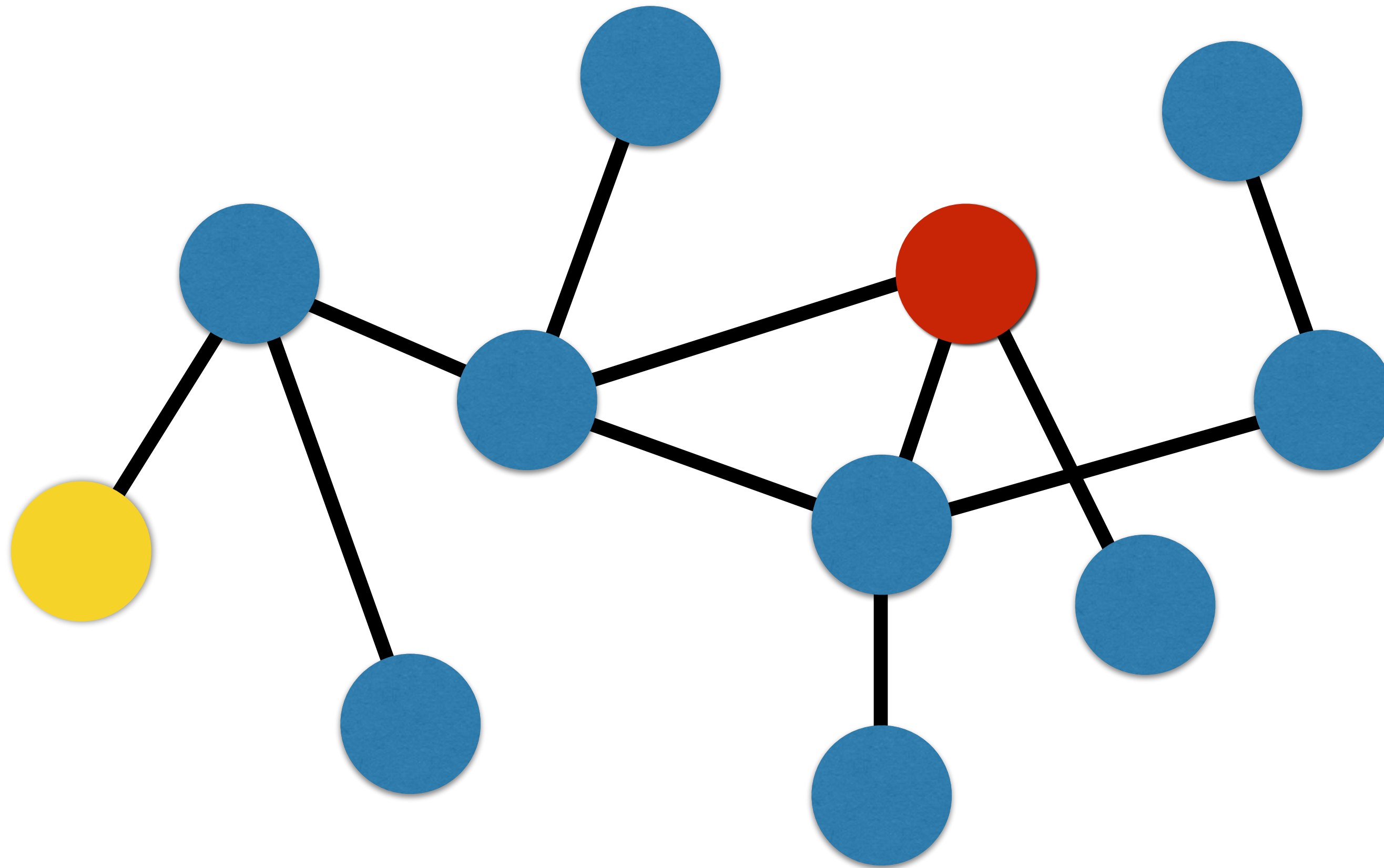
Finding paths

- Pathfinding is important for
 - Optimization: e.g. shortest transport paths
 - Modeling: e.g. disease spread, information passing
- Algorithm: Breadth-first search



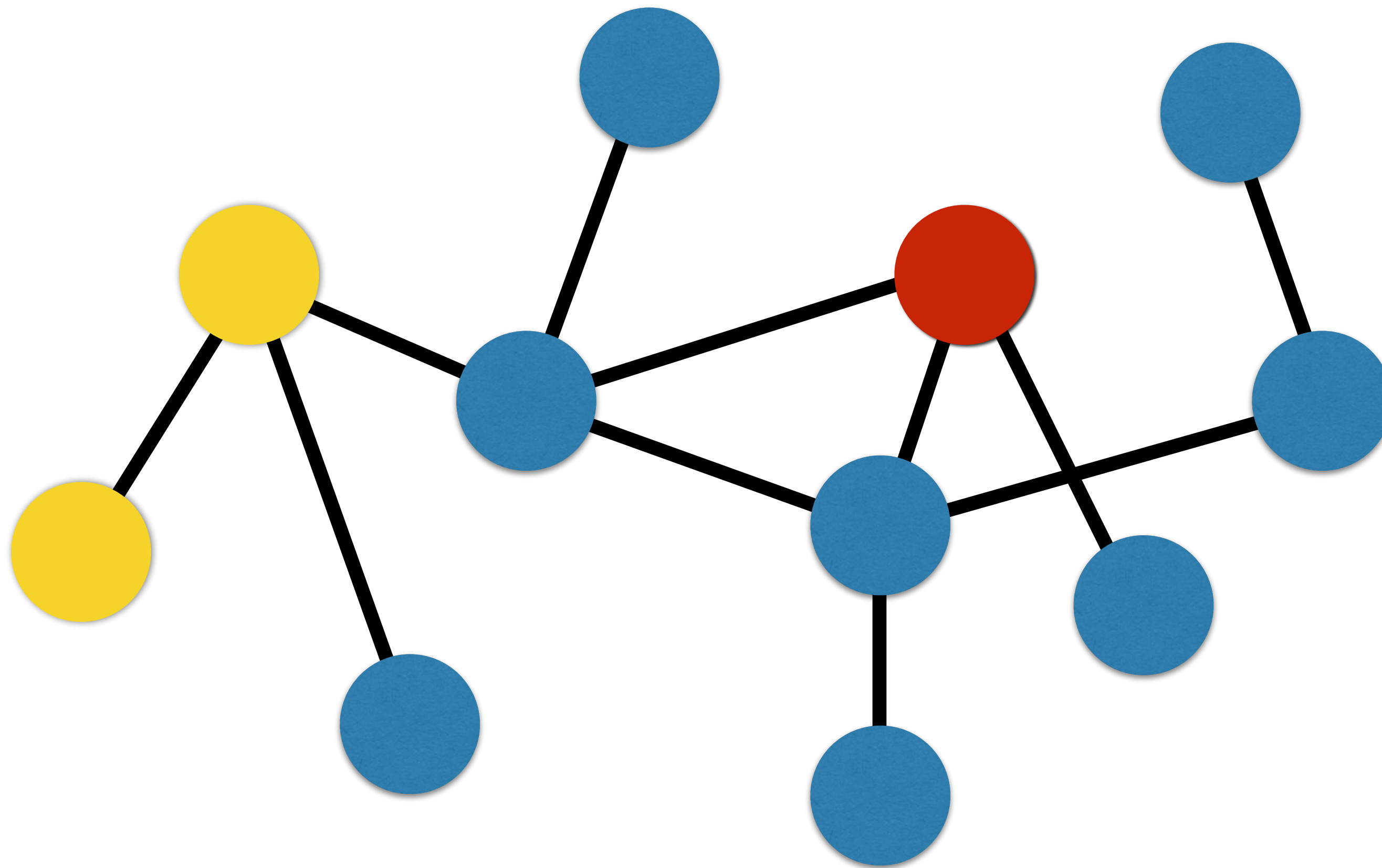
Breadth-first search (BFS)

- Example: Shortest path between two nodes



Breadth-first search (BFS)

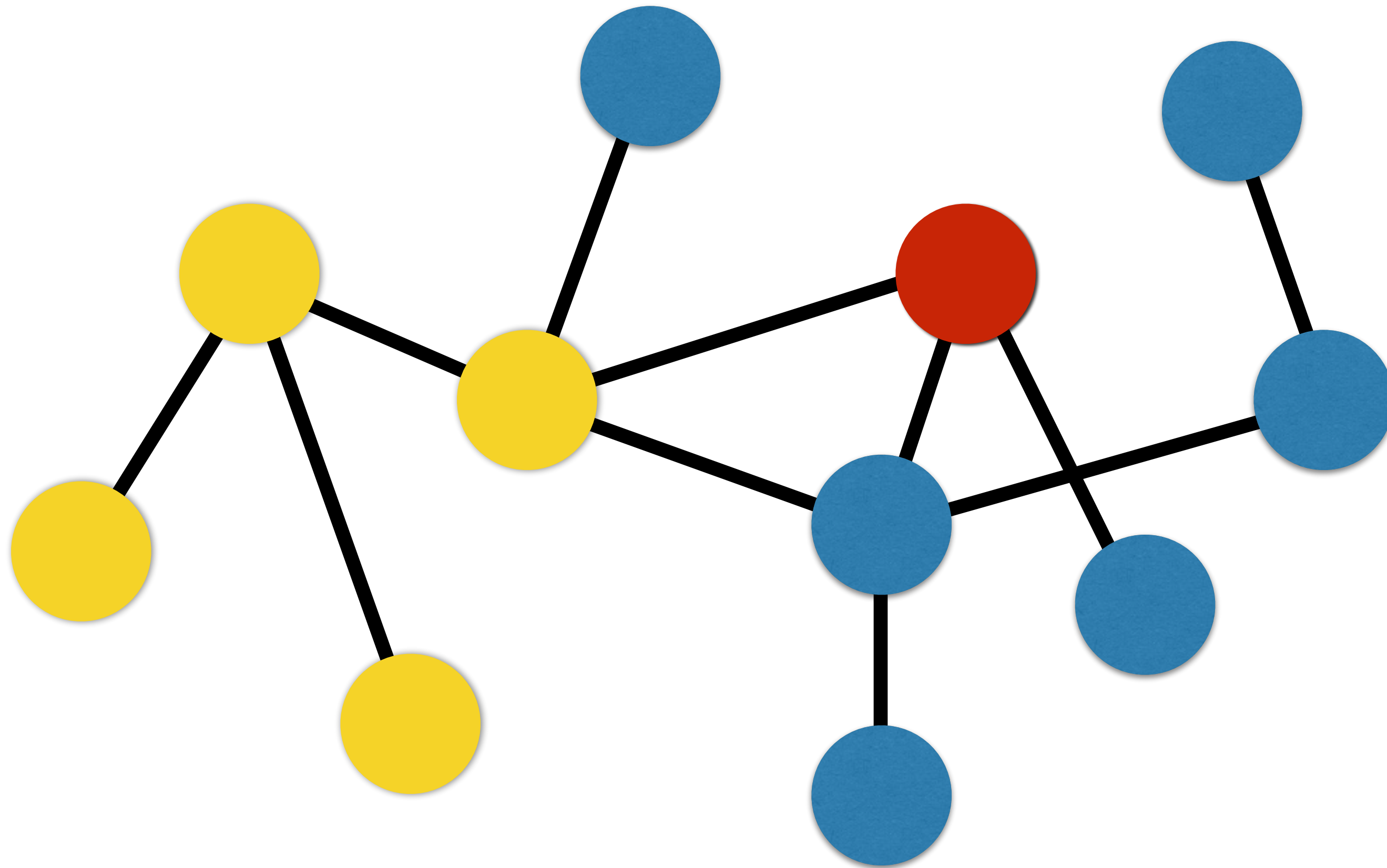
- Example: Shortest path between two nodes





Breadth-first search (BFS)

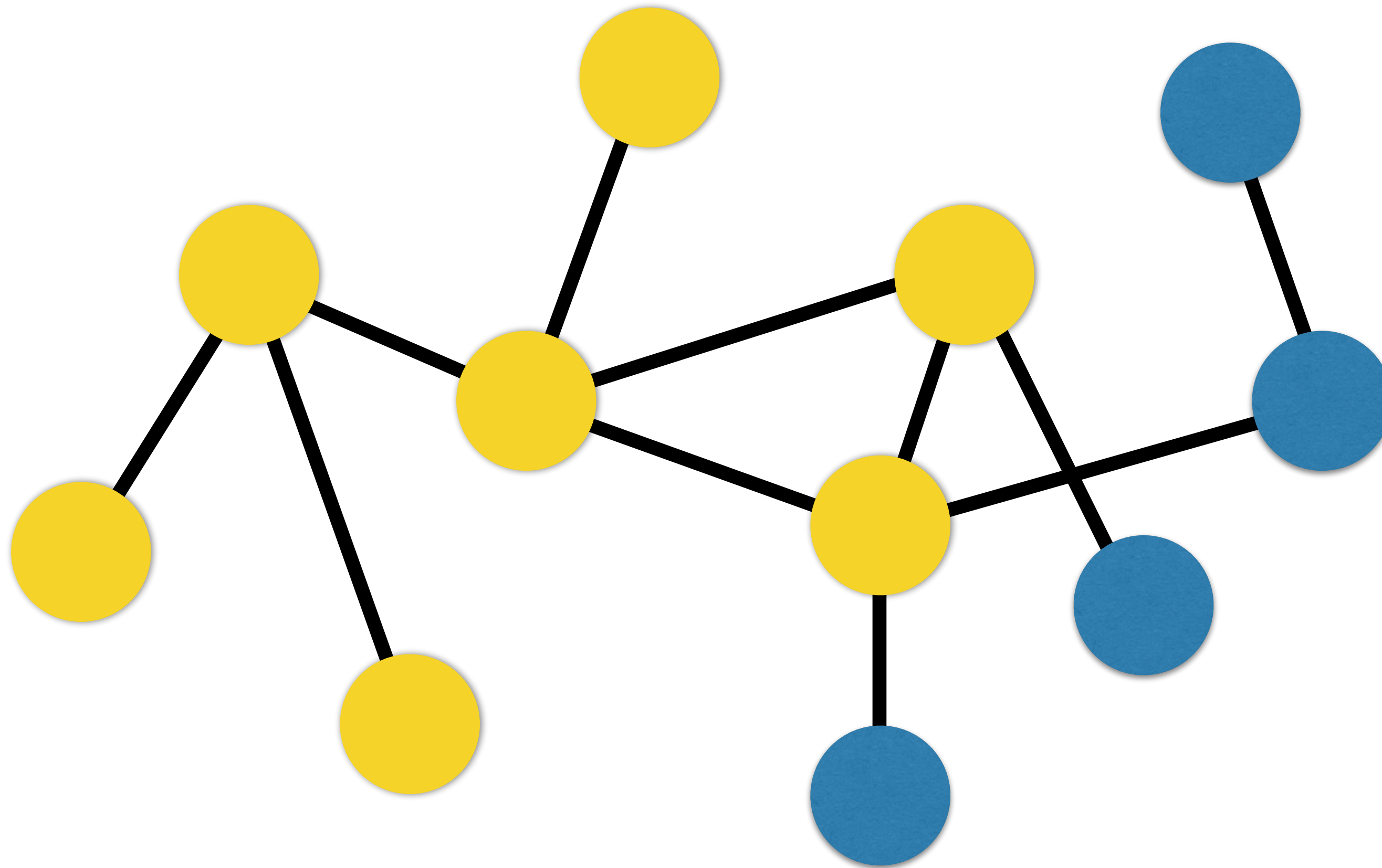
- Example: Shortest path between two nodes





Breadth-first search (BFS)

- Example: Shortest path between two nodes





Recall: Neighbors

```
In [1]: G
Out[1]: <networkx.classes.graph.Graph at 0x10cc08828>
```

```
In [2]: len(G.edges())
Out[2]: 57
```

```
In [3]: len(G.nodes())
Out[3]: 20
```

```
In [4]: G.neighbors(1)
Out[4]: [10, 5, 14, 7]
```

```
In [5]: G.neighbors(10)
Out[5]: [1, 19, 5, 17, 8, 9, 13, 14]
```



NETWORK ANALYSIS IN PYTHON I

Let's practice!



NETWORK ANALYSIS IN PYTHON I

Betweenness centrality



All shortest paths

- Set of paths
- Each path is shortest path between a given pair of nodes
- Done for all node pairs



Betweenness centrality

- Definition:

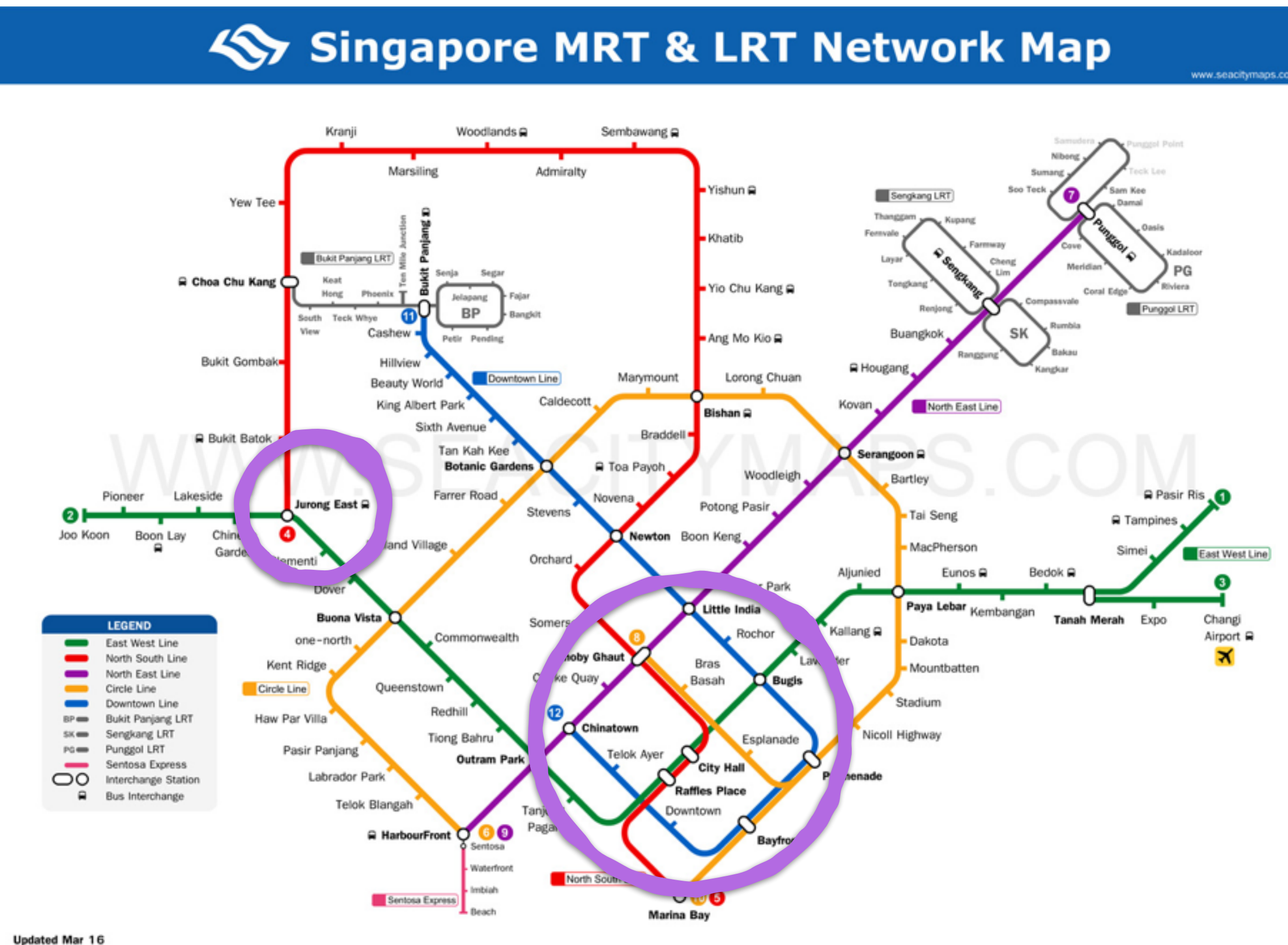
$$\frac{\text{num. shortest paths through node}}{\text{all possible shortest paths}}$$

- Application:

- Bridges between liberal- and conservative-leaning Twitter users
- Critical information transfer links

Examples

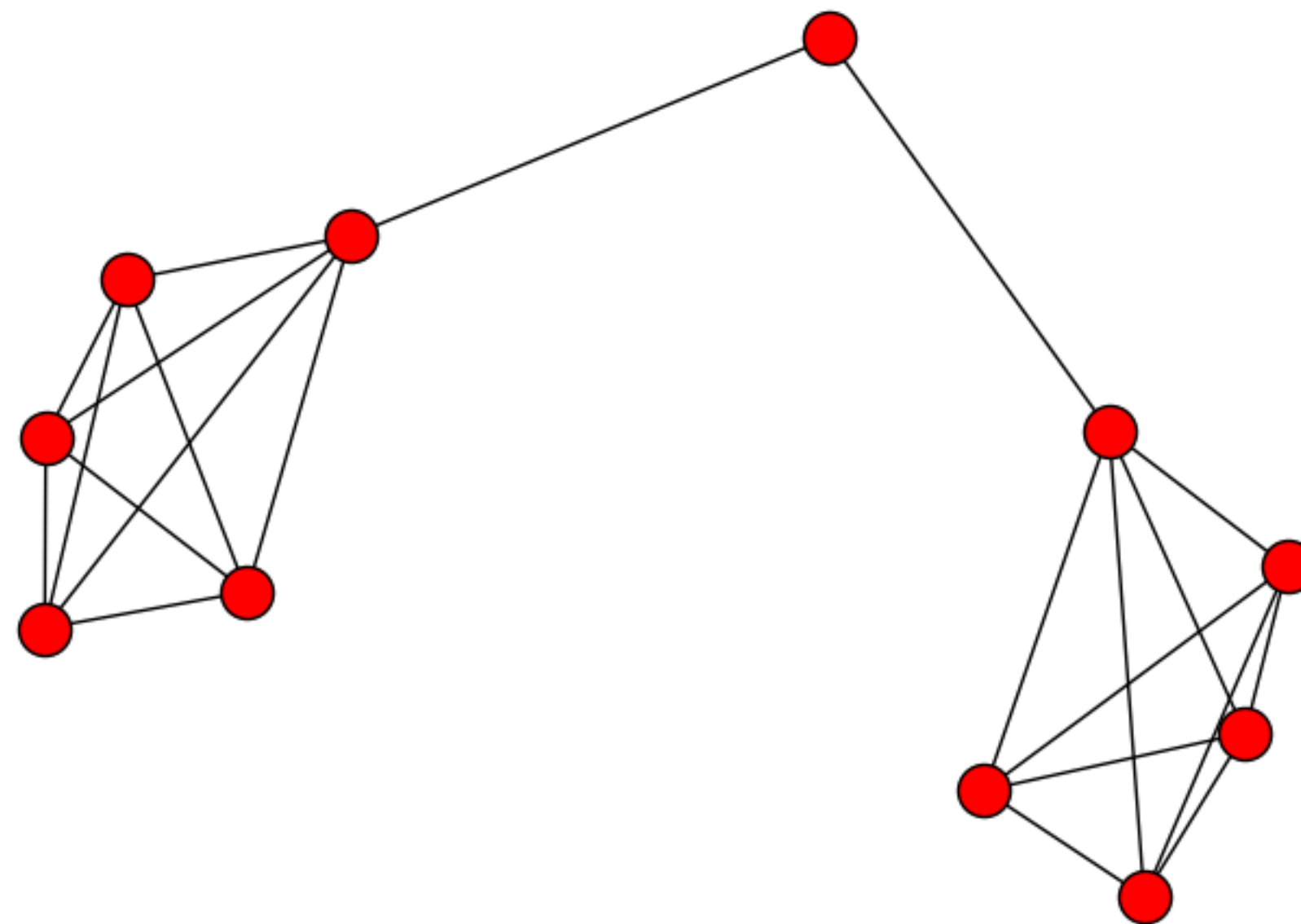
- Singapore: Raffles Place & Jurong East





Example

- High betweenness centrality, low degree centrality?





Betweenness centrality

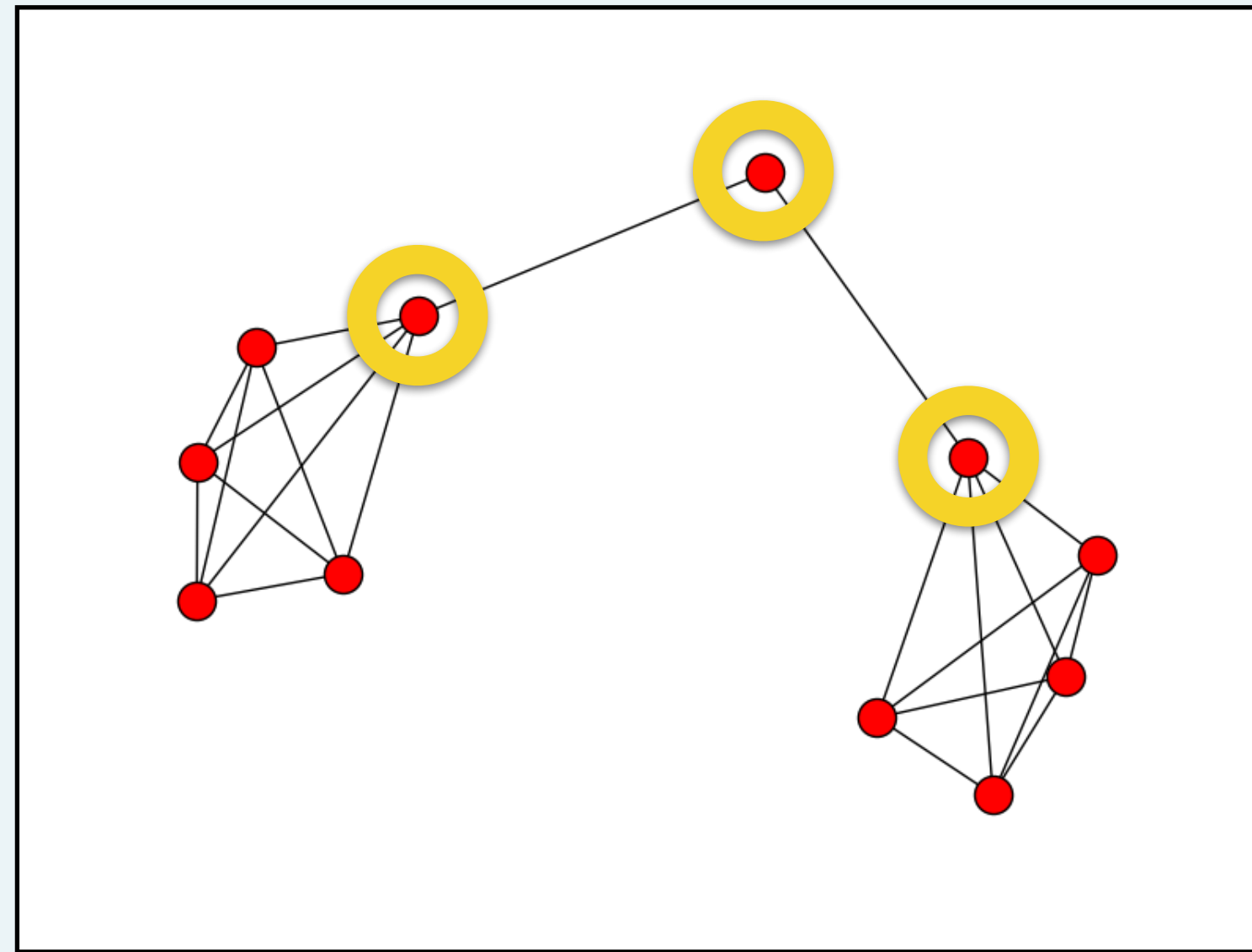
```
In [5]: import networkx as nx
```

```
In [6]: G = nx.barbell_graph(m1=5, m2=1)
```

```
In [10]: nx.betweenness_centrality(G)
```

```
Out[10]:
```

```
{0: 0.0,  
 1: 0.0,  
 2: 0.0,  
 3: 0.0,  
 4: 0.5333333333333333,  
 5: 0.5555555555555556,  
 6: 0.5333333333333333,  
 7: 0.0,  
 8: 0.0,  
 9: 0.0,  
10: 0.0}
```





NETWORK ANALYSIS IN PYTHON I

Let's practice!