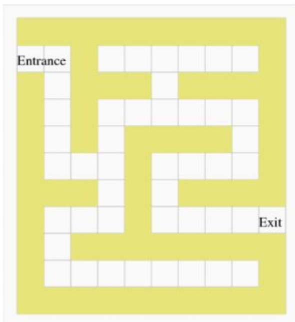
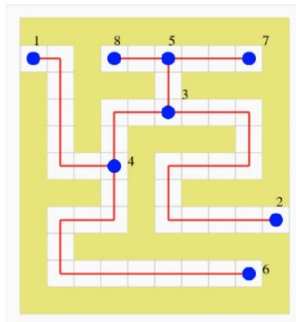
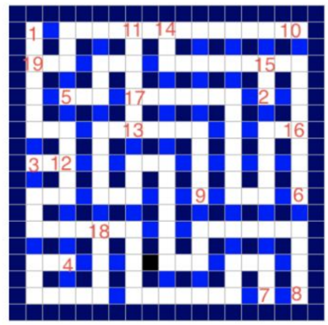
Abelardo Riojas
ISC 4221C Assignment 2
Prof. Quaife, Fall 2019

1. Maze question







Right hand rule node sequence: 1 ⟶ 4 ⟶ 6 ⟶ 4 ⟶ 3 ⟶ 2

```
Edgelist  = [
1, 4;
2, 3;
3, 4;
3, 5;
4, 6;
5, 7;
5, 8 ];
```

```
adjMatrix = [
0 0 0 1 0 0 0 0;
0 0 1 0 0 0 0 0;
0 1 0 1 1 0 0 0;
1 0 1 0 0 1 0 0;
0 0 1 0 0 0 1 1;
0 0 0 1 0 0 0 0;
0 0 0 0 1 0 0 0;
0 0 0 0 1 0 0 0
];
```

Right hand rule sequence for graph 2: 1 ⟶ 19 ⟶ 12 ⟶ 3 ⟶ 12 ⟶ 18 ⟶ 4 ⟶ 18 ⟶ 7 ⟶ 18 ⟶ 12 ⟶ 19 ⟶ 11 ⟶ 5 ⟶ 17 ⟶ 13 ⟶ 8 ⟶ 13 ⟶ 9 ⟶ 13 ⟶ 17 ⟶ 16 ⟶ 6 ⟶ 16 ⟶ 15 ⟶ 14 ⟶ 10 ⟶ 14 ⟶ 11 ⟶ 19 ⟶ 1

Depth first search steps:

Step 0: I am at node 1, I see node 19, and I am about to move to node 19.
Stack: ∅

Step 1: I am at node 19, I see 12 and 11, and I am about to move to node 11.
Stack: 12
Step 2: I am at node 11, I see 14, 17 and 5, and I am about to move to node 14.
Stack: 12, 17, 5
Step 3: I am at node 14, I see 10 and 15, and I am about to move to node 15.
Stack: 12, 17, 5, 10.
Step 4: I am at node 15, I see 2 and 16, and I am about to move to node 2.
Stack: 12, 17, 5, 10, 16
Step 5: I have reached node 2, I am done (and also very lucky).

2. Connectivity Algorithm
Consider the edgelist:

$$\{\{1, 2\}, \{1, 3\}, \{1, 9\}, \{2, 3\}, \{2, 5\}, \{4, 6\}, \{4, 7\}, \{5, 8\}, \{6, 7\}\}.$$

Apply connection algorithm

c = 1	Used New Untouched Component	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0
c = 1	Used New Untouched Component	0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0
c = 1	Used New Untouched Component	1 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 1 0 0 0 1 1 1 1 1 0 1 1 1 0 0 0 0 0 1

c = 2	Used New Untouched Component	1 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 1 0 0 0 1 1 1 1 1 0 1 1 1 0 0 0 0 0 1
c = 2	Used New Untouched Component	1 1 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 1 0 1 1 1 0 1 1 1 0 1 0 0 0 1

c = 3	Used New Untouched Component	1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 1 1 0 1 1 1 0 1 0 0 0 1
c = 3	Used New Untouched Component	1 1 1 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 1 0 1 1 1 0 1 1 1 0 1 0 0 0 1

c = 5	Used New Untouched Component	1 1 1 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 1 1 0 0 1 1 1 0 1 0 0 0 1
c = 5	Used New Untouched Component	1 1 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 1 0 0 1 1 1 0 1 0 0 1 1

New is empty, there is a subgraph.

c = 4	Used New Untouched Component	1 1 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 1 0 0 0 0 0
c = 4	Used New Untouched Component	1 1 1 1 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 1 1 0 0

2 Subgraphs: First one with nodes A, B, C, E, H, I, the second one with nodes D, F, G.

3. Graphs

Dijkstra's Algorithm:

Step	P	Connect	Dist
0	A	0 0 0 0 0 0 0 0 0 0 0 0 0	∞ ∞ ∞ ∞ ∞ ∞ ∞ ∞ ∞ ∞ ∞ ∞ ∞

1	A	1 0 0 0 0 0 0 0 0 0 0 0 0	0 ∞ ∞ ∞ ∞ ∞ ∞ ∞ ∞ ∞ ∞ ∞ ∞

For every unconnected node with a finite edge, (2, 6, and 7)
Dist(2) = min(dist(2), dist(1)+Length(2,1))
Dist(2) = min(∞, 0 + 2.23)
Dist(2) = 2.23.

Dist(6) = min(dist(6), dist(1) + Length(6,1))
Dist(6) = min(∞, 0 + 6.00)
Dist(6) = 6.00.

Dist(7) = min(dist(7), dist(1) + Length(7,1))
Dist(7) = min(∞, 0 + 5.38)
Dist(7) = 5.38.

Set P to the shortest unconnected node (2). Check is P = L (No.)

Step	P	Connect	Dist
2	B	1 1 0 0 0 0 0 0 0 0 0 0 0	0 2.23 ∞ ∞ ∞ 6.00 5.38 ∞ ∞ ∞ ∞ ∞ ∞

For every unconnected node with a finite edge, (3, 4, and 5)
Dist(3) = min(dist(3), dist(2)+Length(3,2))
Dist(3) = min(∞, 2.23 + 2.00)
Dist(3) = 4.23.

Dist(4) = min(dist(4), dist(2)+Length(4,2))
Dist(4) = min(∞, 2.23 + 2.00)
Dist(4) = 4.23.

Dist(5) = min(dist(5), dist(2)+Length(5,2))
Dist(5) = min(∞, 2.23 + 2.83)
Dist(5) = 5.06.

Set P to the shortest unconnected node (tie between 3 and 4, picked 4).
Check if P = L (No.)

Step	P	Connect	Dist
3	D	1 1 0 1 0 0 0 0 0 0 0 0 0	0 2.23 4.23 4.23 5.06 6.00 5.38 ∞ ∞ ∞ ∞ ∞ ∞

For every unconnected node with a finite edge, (5 and 6)
Dist(5) = min(dist(5), dist(4)+Length(5,4))
Dist(5) = min(5.06, 4.23 + 2.00)
Dist(5) = 5.06.

Dist(6) = min(dist(6), dist(4)+Length(6,4))
Dist(6) = min(6.00, 4.23 + 2.23)
Dist(6) = 6.00

Set P to the shortest unconnected node (5). Check if P = L (No.)

Step	P	Connect	Dist
4	E	1 1 0 1 1 0 0 0 0 0 0 0 0	0 2.23 4.23 4.23 5.06 6.00 5.38 ∞ ∞ ∞ ∞ ∞ ∞

For every unconnected node with a finite edge, (3, 7 and 12).

Dist(3) = min(dist(3), dist(5)+Length(3,5))
Dist(3) = min(4.23, 5.06 + 2.00)
Dist(3) = 4.23.

Dist(7) = min(dist(7), dist(5)+Length(7,5))
Dist(7) = min(5.38, dist(5.06 + 2.83)
Dist(7) = 5.38.

Dist(12) = min(dist(12), dist(5) + Length(12,5))
Dist(12) = min(∞, 5.06 + 4.47)
Dist(12) = min(9.53).

Set P = L. We are done. Length of shortest path = 9.53.

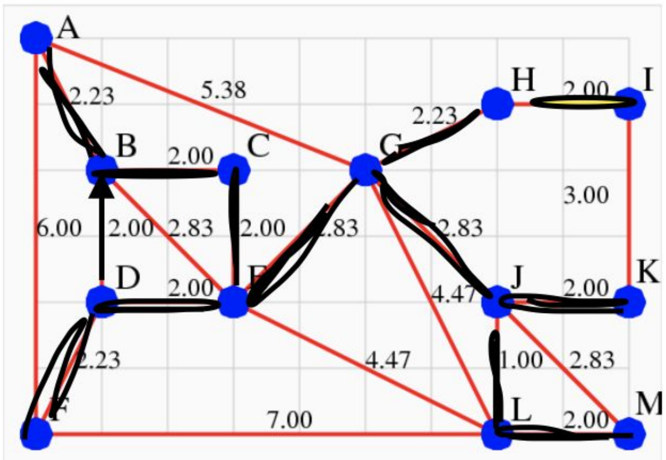
Minimum Spanning Tree

Need 12 edges, keep adding smallest edges that do not make a cycle.

Add edge: 10, 12 (Length 1.00). No cycle.
Add edge: 2, 3 (Length 2.00). No cycle.
Add edge: 2, 4 (Length 2.00). No cycle.
Add edge: 3, 5 (Length 2.00). No cycle.
Add edge: 4, 5 (Length 2.00). No cycle.
Add edge: 8, 9 (Length 2.00). No cycle.
Add edge: 10, 11 (Length 2.00) No cycle.
Add edge: 12, 13 (Length 2.00) No cycle.

```
Add edge: 1, 2 (Length 2.23) No cycle.
Add edge: 4, 6 (Length 2.23) No cycle.
Add edge: 7, 8 (Length 2.23) No cycle.
Add edge: 5, 7 (Length 2.83) No cycle.

Total length = 24.52
```

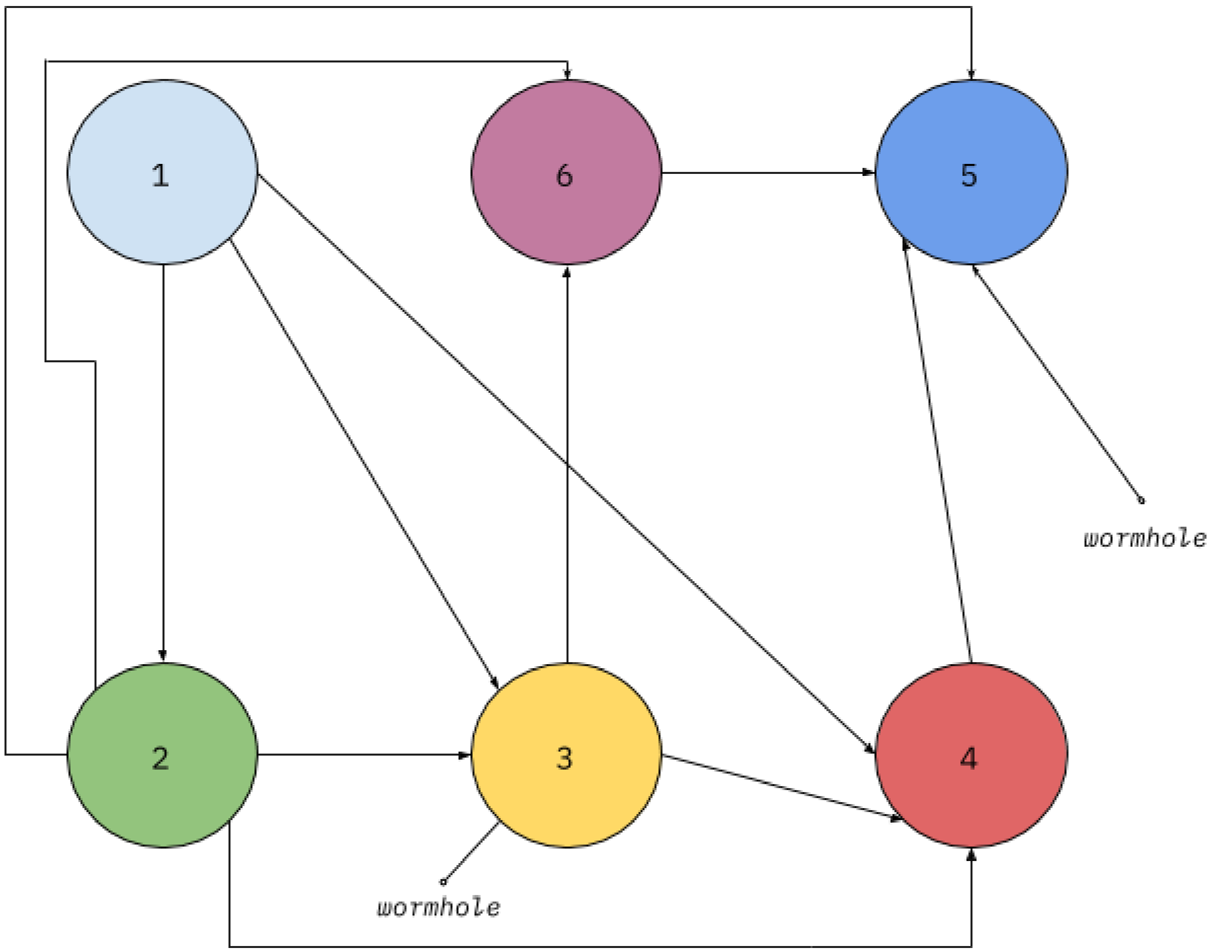


4. 6 Degrees of Kevin Bacon

Nodes

1. Kevin Bacon
2. Robert DeNiro
3. Jack Nicholson
4. Meryl Streep
5. Diane Keaton
6. Susan Saradon

```
Edgelist = [
      1,2;
      1,3;
      1,4;
      2,3;
      2,4;
      2,5;
      2,6;
      3,4;
      3,5;
      3,6;
      4,5;
      5,6 ];
```



The shortest length for any pair of actors that haven't been in a movie together is 3. Any actor in this list can go through DeNiro or Nicholson to get to any other actor since those two have acted with everyone on the list.

Bacon Keaton paths:

- Bacon ⟶ DeNiro ⟶ Keaton
- Bacon ⟶ Nicholson ⟶ Keaton
- Bacon ⟶ Streep ⟶ Keaton

Bacon Sarandon paths:

- Bacon ⟶ DeNiro ⟶ Sarandon

Bacon → Nicholson → Sarandon

Sarandon Streep paths:

Sarandon → DeNiro → Streep
Sarandon → Nicholson → Streep
Sarandon → Keaton → Streep

Bacon centrality:
Shortest path between Bacon and DeNiro, Nicholson, and Streep do not go through a middle point *v.* (3)
Bacon is not the middle man for any two other unconnected nodes.

Nicholson centrality:
Shortest path between Nicholson and any actor does not go through a middle point *v.* (5)
Nicholson is the middleman for all 3 shortest-unconnected nodes paths. (3)

DeNiro centrality:
Shortest path between DeNiro and any actor does not go through a middle point *v.* (5)
DeNiro is the middleman for all 3 shortest-unconnected nodes paths. (3)

Streep centrality:
Shortest path between Streep and Bacon, DeNiro, and Nicholson does not go through a middle point *v.* (3)
Streep is the middle man for one shortest-unconnected nodes path. (1)

Sarandon centrality:
Shortest path between Sarandon and Nicholson, DeNiro does not go through a middle point *v.* (2)
Sarandon is not the middle man for any two other unconnected nodes.

Ranking:

DeNiro/Nicholson
Streep
Bacon
Sarandon