Project 2

Name: Ameer Hassan

Section: 004

Github URL: https://github.com/AmeerYHassan/CS435-Project-Two

Name of ALL collaborators: No collaborators to list.

URLs/ISBNs for ALL consulted websites/textbooks:

- https://www.youtube.com/watch?v=HDUzBEG1GlA
- https://www.youtube.com/watch?v=QVcsSaGeSH0
- In-Class notes and slides for some help with the pseudocode.

1. Gonna Take My Horse to the Old Town Node

a. Breadth First Search: S → A → C → B → E → F → G → K → L → D

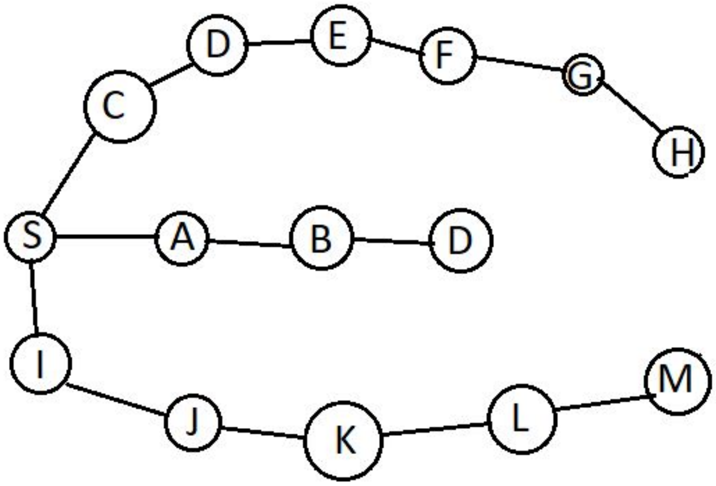
b. Adjacency Matrix

	S	**A**	**C**	**B**	**E**	**F**	**G**	**K**	**L**	**D**
S	0	1	0	0	0	0	0	0	0	0
A	1	0	1	1	1	0	0	0	0	0
C	0	1	0	0	0	0	0	0	0	0
B	0	1	0	0	1	0	0	0	0	0
E	0	1	0	1	0	1	0	0	0	0
F	0	0	0	0	1	0	1	0	0	0
G	0	0	0	0	0	1	0	1	0	0
K	0	0	0	0	0	0	1	0	1	0
L	0	0	0	0	0	0	0	1	0	1
D	0	0	0	0	0	0	0	0	1	0

c. Draw a graph where DFS beats BFS

i. BFS: S → C → A → I → D → B → J → E → D

ii. DFS: S → A → B → D



2. Boulevard of Broken Cheese

a. If the maze was converted to a graph, it would take 25 nodes, one for each square

b. The edges of the graph would represent the possible route the mouse could take. If two squares in the graph have a wall between them, then there can’t be an edge in the graph representation.

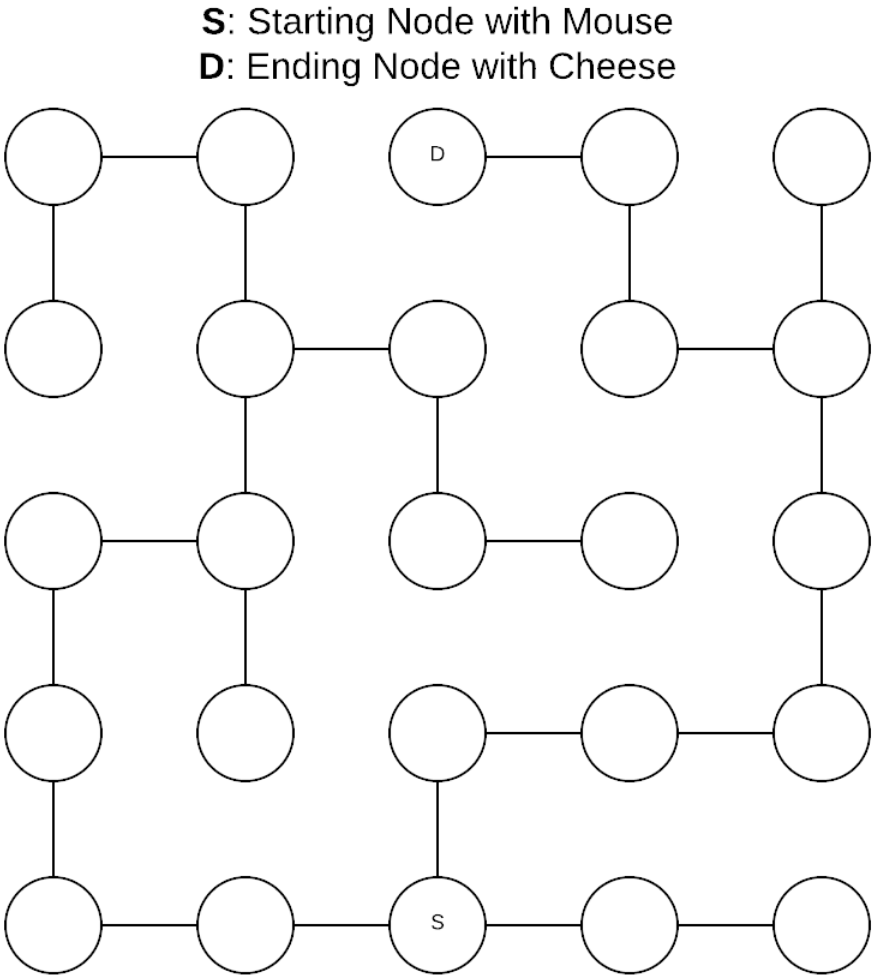
c. Properties of the graph:

   i. Undirected Edges

   ii. Connected

   iii. Unweighted

d. Graph Representation:


S: Starting Node with Mouse
D: Ending Node with Cheese
D
S

3. Traverse This Town

a. Why would BFTRec fail on a linked list with 10,000 nodes?

    i. This did fail for me, and this is because of the recursion depth. If the recursion depth becomes too high, then the program can't keep up and crashes. This can be fixed if you set the recursion depth to some arbitrarily high number, but this question shows how recursion is not very suitable for a traversal.

b. Why would BFTIter work?

    i. BFTIter would work as long as there is enough memory. Since it's iterative and does not use recursion, it does not need to worry about the recursion depth limit.

4. Thank U, Vertext

a. A DAG, or Directed Acyclic Graph, is a directed graph (edges have arrows and can only go one way), and is a graph that has no cycles. Compared to the generation of a graph in the previous part, this graph has to make sure there are no cycles, and when you add an edge between two nodes, they must have a direction towards them.

5. Uno, Do', Tre', Cuatro, I Node You Want Me

a. The properties of the graph that make it possible to use Djikstra's algorithm is the fact that there are "treadmills" to the graph, adding weight to the edges. Since I am a lazy mouse, Djikstra's algorithm would be the optimal solution to this since it minimizes the amount of power I have to work against the treadmill with.

6. When You Wish Upon A*

a. An admissible heuristic for this problem is to use the Manhattan distance. With the Manhattan distance, the value returned will NEVER overestimate, making it an admissible heuristic.