Team members: Brock Weekley (I reached out to the members of my initial group and no one responded)

Uniform Cost Tree Search: The uniform cost tree search is implemented by adding the initial node to the fringe. The lowest cumulative cost node is expanded and each touching room is added to the fringe in the order: Down, Up, Right, Left (Lowest cost order). This operation is looped until the fringe is empty. This means that the same node can be expanded multiple times, and the lowest cost path will be found.

Uniform Cost Graph Search: The uniform cost graph search is implemented in the same way as the uniform cost tree search, with the addition of a closed list. When a node is expanded, its neighbors are only added if it is in bounds AND not found in the closed list. This means that the same node cannot be expanded multiple times, and the lowest cost path will be found.

Iterative Deepening Tree Search: The iterative deepening tree search is implemented by adding the initial node to the fringe. The first item in the fringe is expanded and each touching room is added to the fringe in the order: Up, Left, Right, Down (Lowest row #, lowest column # order). This operation is looped until the depth limit has been reached. The depth limit is then increased and the operation is run again. The depth limit is determined by the size of the previous fringe (i.e. If the fringe had four nodes, increasing the depth limit by one means looping four more times in order to expand each of those nodes). This operation also ends when all of the rooms are clean. This means that the same node can be expanded multiple times.

Programming language: Python
Hardware: Windows 10 PC

Execution Results:

UCTS #1: First five expanded nodes, instance one: (2, 2), (3, 2), (1, 2), (2, 3), (2, 1)
Generated nodes: 600
Expanded nodes: 178
Execution time: 0.017s - 0.024s
Path found: (2, 2), (1, 2), (2, 2), (2, 3), (2, 4), (3, 4), (3, 5)

UCTS #2: First five expanded nodes, instance two: (3, 2), (4, 2), (2, 2), (3, 3), (3, 1)
Generated nodes: 186
Expanded nodes: 54
Execution time: 0.0009s - 0.002s
Path found: (3, 2), (2, 2), (1, 2), (1, 3), (2, 3), (3, 3), (4, 3), (4, 4), (3, 4), (2, 4)

UCGS #1: First five expanded nodes, instance one: (2, 2), (3, 2), (1, 2), (2, 3), (2, 1)
Generated nodes: 62
Expanded nodes: 43
Execution time: 0.0009s - 0.001s

Path found: (2, 2), (1, 2), (1, 3), (2, 3), (2, 4), (3, 4), (3, 5)

UCGS #2: First five expanded nodes, instance two: (3, 2), (4, 2), (2, 2), (3, 3), (3, 1)
Generated nodes: 50
Expanded nodes: 28
Execution time: 0.0009s - 0.001s
Path found: (3, 2), (2, 2), (1, 2), (1, 3), (2, 3), (3, 3), (4, 3), (4, 4), (3, 4), (2, 4)

IDTS #1: First five expanded nodes, instance one: (2, 2), (2, 2), (1, 2), (2, 1), (2, 3)
Generated nodes: 716
Expanded nodes: 323
Max Depth: 5
Execution time: 0.011s - 0.022s
Path found: (2, 2), (1, 2), (1, 3), (1, 4), (2, 4), (2, 5), (3, 5)

IDTS #2: First five expanded nodes, instance two: (3, 2), (3, 2), (2, 2), (3, 1), (3, 3)
Generated nodes: 283
Expanded nodes: 119
Max Depth: 4
Execution time: 0.006s - 0.012s
Path found: (3, 2), (2, 2), (1, 2), (1, 3), (1, 4), (2, 4), (2, 3), (3, 3), (3, 4), (4, 4)