Problem Wk.13.3.3: Paths in the Map

Look at `tutor13Work.py`. Call the `search.search` function to find paths between the specified nodes in `map1` using the four search methods (breadth-first and depth-first, with and without dynamic programming). The documentation for `search.search` can be found in Section 8.1 of the class notes and in the Software Documentation for the `search` module.

The `search.search` function takes an `actions` argument which should be a list of legal actions. The `successor` function is called with a state and each of the actions on the list. If an action is not legal in a particular state, the successor function should just return the input state.

Make sure that you set `search.verbose = True`.

Look at the number of states visited as well as the paths and make sure that you understand the differences. You can set `search.verbose` to `True` to see every step in the search in detail.

1. **Assume the start state is A and the goal state is G.** Enter the path found by breadth-first search without dynamic programming. Enter a sequence of state names, e.g. ACD.

2. How many states were visited during the search?

3. How many nodes were expanded during the search?

4. **Assume the start state is A and the goal state is G.** Enter the path found by breadth-first search with dynamic programming. Enter a sequence of state names, e.g. ACD.

5. How many states were visited during the search?

6. How many nodes were expanded during the search?

7. Enter the name of a state the was visited more than once by breadth-first without DP.

8. The path found by breadth-first search with and without DP should generally be the same path. Enter True or False.

9. Enter the maximum number of states that can be visited by ANY breadth-first search with DP in `map1` (start of path is not counted as visited).

10. **Assume the start state is G and the goal state is C.** Enter the path found by depth-first search without dynamic programming. Enter a sequence of state names, e.g. ACD.

11. How many states were visited during the search?

12. How many nodes were expanded during the search?

13. **Assume the start state is G and the goal state is C.** Enter the path found by depth-first search with dynamic programming. Enter a sequence of state names, e.g. ACD.

14. How many states were visited during the search?

15. How many nodes were expanded during the search?

16. Enter the name of a state the was visited more than once by depth-first without DP.

17. Enter the maximum number of states that can be visited by ANY depth-first search with DP in `map1` (start of path is not counted as visited).

18. The path found by depth-first search with and without DP should generally be the
same path. Enter True or False.