Question 2

3. (5 points) ***Sept 12*** **Run this with the included BFS and DFS implementations. Extend each of these to count the number of states generated. Print this out at the end**.

BFS - 3
DFS - 3
Goal not found for both

4. (10 points) ***Sept 13*** **Extend the depth_first_search function to implement** ***depth_limited_search*** **by providing an optional** ***limit*** **parameter. When you are generating successors, only go to depth=limit in the search tree. (You are welcome to extend the RoverState class to make this easier if you'd like.)**

DLS - 3 with limit = 7

5. Run each of the three algorithms (breadth_first_search, depth_first_search, depth_limited_search) on this new problem and count the number of states generated.

BFS - 19
DFS - 14
DLS - 14 with limit = 7

6. (10 points) ***Sept 15*** **An early insight in solving search-based problems was the idea of** ***problem decomposition*****. If a problem can be subdivided into smaller components that can be solved separately, we can deal with scaling more easily. Modify your search code so that it instead solves three subproblems: moveToSample, removeSample, and returnToCharger. You can do this by changing the start state and goal test. How does this change the number of states generated?**

(with an action list specific for each subproblems)
Total states generated by decomposition for BFS = 12
Total states generated by decomposition for DFS = 12
Total states generated by decomposition for DLS = 12

Question 3

c) (5 points) ***Sept 18*** **Run both A* and uniform cost search (i.e. using h1: h=0 for all states) on the MarsMap and count the number of states generated. Add this to your results**

Total states generated for UCS: 32
Total states generated for A*: 32