1. What are two potential pitfalls of using recursion? How could you overcome these challenges (either by using an alternative technique or by doing something to help the recursion)?

Firstly, it is important that when using recursion, the function is called for solving a smaller problem. If not, the function will run forever and cause a stack overflow, because the function will keep on calling a case where it requires another function call, which lengthens the run time. Thus, we have to make sure that the recursive call does not call on to solve a larger problem. Recursion can still be used but with memorization. When performing memorization, the function has a dictionary to store intermediate results of the function, creating new base cases, and allowing the function to use the results stored in the dictionary, without having to recursively call the function unnecessarily.

Secondly, it is important to set a base case, which exits the function. Since with recursion, the function is called over and over again at the end of the function, without a base case, the code will be trapped in the function forever, causing the code to crash by running the function infinite times. A base case means that this case does not require any recursion, the function itself already is able to return the value or action needed. Thus, this marks the end of the function. Recursion will end in that case.