1. What would be the best data structure to model the behaviour of the supply basket? Justify your answer and explain how it would be used.
2. What is the time complexity of adding new seeds to the supply basket?
3. What is the time complexity of removing seeds from the supply basket?
4. What is the time complexity of checking to see if the supply basket is empty?

Response:

Q1:

A stack would be the best option for the game. A stack adopts the “Last In, First Out” method while pushing and popping items. According to the instructions, the player can only plant the seed that they most recently picked up, since for a stack, only the last item can be removed. Thus, a stack would be the best data structure for the game.

Q2:

The time complexity of adding new seeds to the supply basket would be O(1). Adding new seeds is a constant-time operation, which does not need any loops, and it does not depend on the sample size, in this case, it is the number of items in the supply basket.

Q3:

The time complexity of removing seeds from the supply basket would be O(1). Removing seeds is a constant-time operation. As the last item is removed, other items will not be shifted to a new index. Only the last item would be operated. Thus, this action is independent of the size of the supply basket.

Q4:

The time complexity of checking if the supply basket is empty would be O(1). The class for the stack will keep track of how many items are currently in there, so no loop is required to obtain the size of the stack. Another constant-time operation is also performed to check if the value obtained is 0, which in the end, gives us O(1).