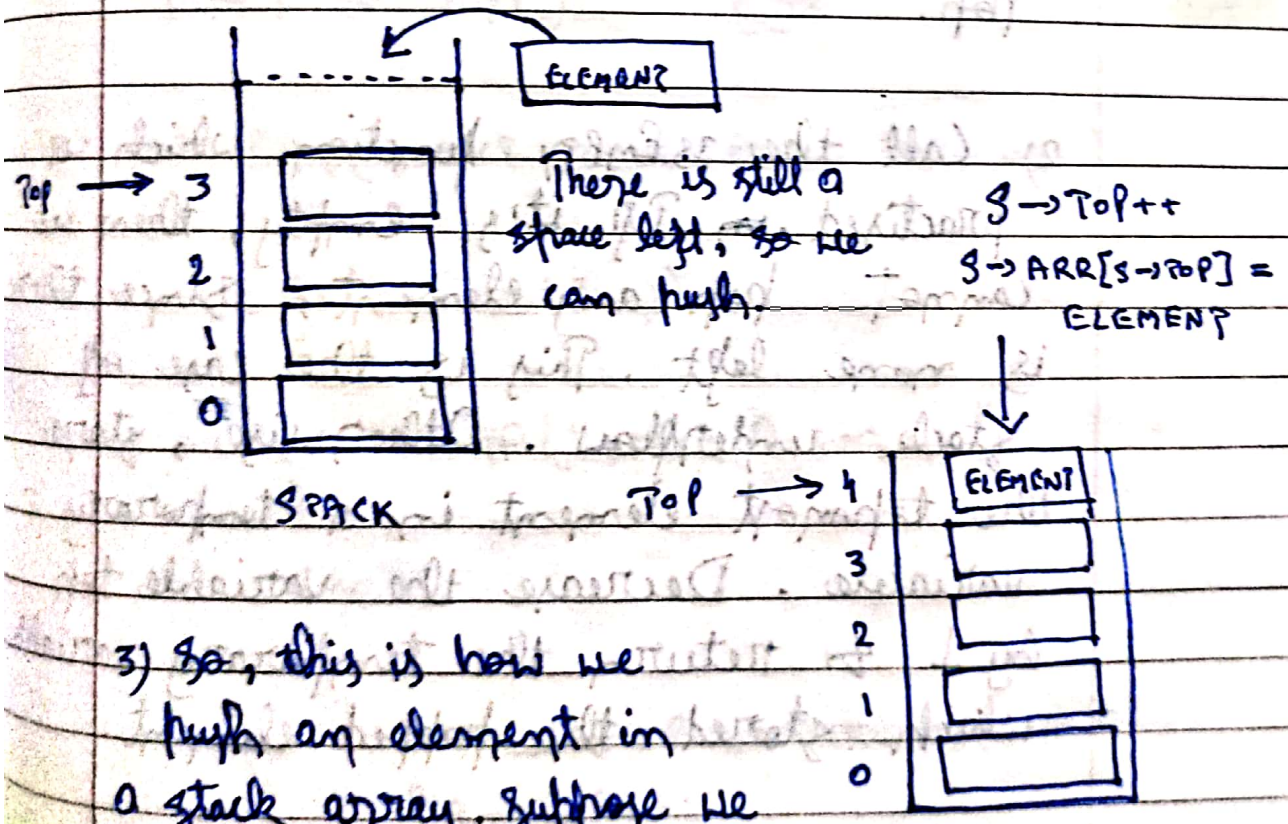
Push() & Pop()

→ Operation 1: PUSH:

1) The first thing is to define a stack. Suppose we have the creating stack and declaring its fundamentals part done.

then pushing an element requires you to first check if there is any space left in the stack.

2) Call the ISFULL function. If it's full, then we cannot push anymore elements. This is the case of stack overflow. Otherwise, increase the variable top by 1 and insert the element at the index top of the stack.



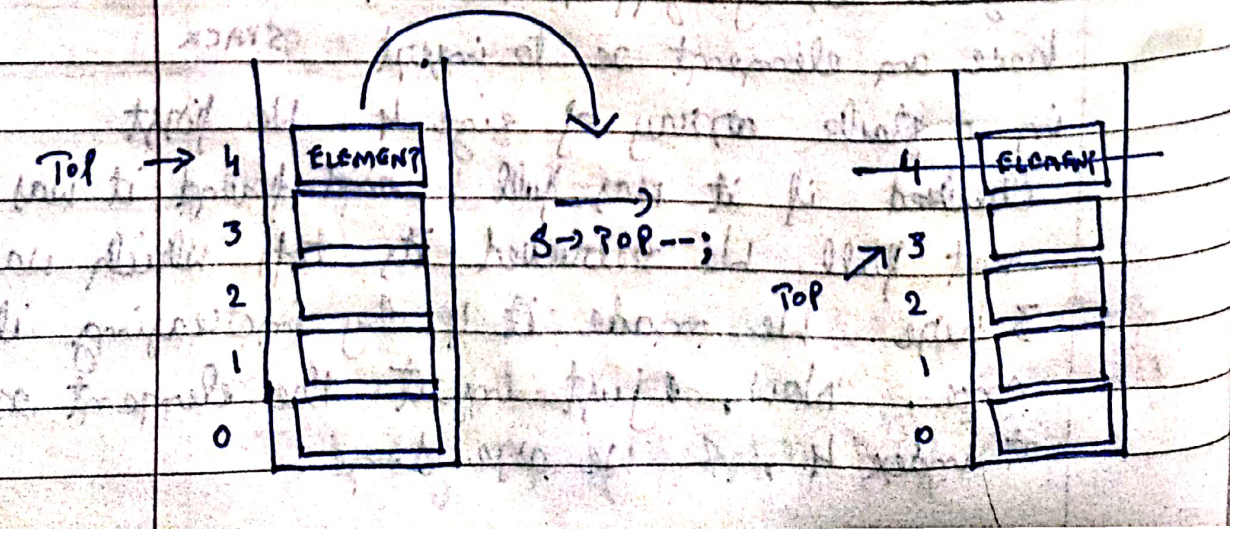
3) So, this is how we push an element in a stack array. Suppose we have an element x to insert in a stack array of size 4. We first checked if it was full, and found it was not full. We retrieved its top which was 3 here. We made it 4 by increasing it once. Now, I just insert the element x at index 4, & we are done.

→ Operation 2: Pop:

→ Popping an element is very similar to inserting an element. You should

1) The first thing again is to define a stack. Get over will all the fundamentals. You must have learnt that by now. Then popping an element requires you to first check if there is any element left in the stack to pop.

2) Call the isEmpty function which we practised. If it's empty, then we cannot pop any element, since there is none left. This is the case of stack underflow. Otherwise, store the topmost element in a temporary variable. Decrease the variable top by 1 & return the temporary variable which stored the popped element.



3) So, this is how we pop an element from a stack array. Suppose we make a pop call in a stack array of size 4. We first checked if it was empty, and found it was not empty. We retrieved its top which was 4 here. Stored the element at 4. We made it 3 by decreasing it once. Now, just returns the element x, and popping is done.