1. The implementation achieves O(1) time for each of these API because there are no loops being used and only one argument. Therefore the runtime will simply be O(1) for each of the functions.

Code snippets

def remove(self,item):

return self.stack.pop(item)

def getValue(self,index):

if index > len(self.stack):

return -1

else:

return self.stack.getitem(index)

As you can see the only argument for remove is item and the only argument for getValue is index.

2. Add achieves O(N) time because it returns the item after adding it to the stack. Therefore it will return depending on where in the stack the item is, which would be O(N) time complexity.

def add(self,item):

return self.stack.append(item)