

Problem 2B

1. My implementation achieves $O(N)$ for the remove function:

```
def remove(self):
    temp = self.data[self.size-1]
    self.size -= 1
    array = []
    for i in range(0, self.size):    //runs n times, hence  $O(N)$ 
        array.append(self.data[i])
    self.data = array
    print(temp)
    return temp
```

My implementation achieves $O(1)$ for the getValue(i) function:

```
def getValue(self, index):
    if index > self.size-1:
        return -1
    else:
        return self.data[index] // This returns the value of index
```

This is because it uses an array, so searching for an index in the array takes $O(1)$ constant time, therefore the time it takes to getValue() is $O(N)$.

2. For the add() function, the time complexity is $O(N)$, because I used the append function to add the element to the end of the array, which takes $O(N)$ time to complete.

```
def add(self, val):
    self.data.append(val) // This appends the value to the end of the array of size n, so  $O(n)$ 
    self.size += 1
```