Custom List Implementation (from scratch)

Features we'll support

- append
- insert
- remove
- pop
- get / set
- size
- display

```python
class MyList:
    def __init__(self):
        self.capacity = 4          # initial capacity
        self.size = 0              # number of elements
        self.data = [None] * self.capacity

    # resize the array when capacity is full
    def _resize(self):
        new_capacity = self.capacity * 2
        new_data = [None] * new_capacity

        for i in range(self.size):
            new_data[i] = self.data[i]

        self.data = new_data
        self.capacity = new_capacity

    # append element at end
    def append(self, value):
        if self.size == self.capacity:
            self._resize()

        self.data[self.size] = value
        self.size += 1
    # insert element at index
    def insert(self, index, value):
        if index < 0 or index > self.size:
            print("Index out of range")
            return

        if self.size == self.capacity:
            self._resize()

        for i in range(self.size, index, -1):
            self.data[i] = self.data[i - 1]

        self.data[index] = value
        self.size += 1

    # remove first occurrence of value
    def remove(self, value):
        for i in range(self.size):
            if self.data[i] == value:
                for j in range(i, self.size - 1):
                    self.data[j] = self.data[j + 1]
                self.data[self.size - 1] = None
                self.size -= 1
                return
        print("Value not found")
```

✦ Get Plus ×

Copy code

```python
# pop element from index (default las
def pop(self, index=None):
    if self.size == 0:
        print("List is empty")
        return

    if index is None:
        index = self.size - 1

    if index < 0 or index >= self.size:
        print("Index out of range")
        return

    value = self.data[index]

    for i in range(index, self.size - 1):
        self.data[i] = self.data[i + 1]

    self.data[self.size - 1] = None
    self.size -= 1
    return value

# get element
def get(self, index):
    if index < 0 or index >= self.size:
        print("Index out of range")
        return
    return self.data[index]
```

```python
# set element
def set(self, index, value):
    if index < 0 or index >= self.size:
        print("Index out of range")
        return
    self.data[index] = value

# display list
def display(self):
    print([self.data[i] for i in range(self.size)])
```

```
lst = MyList()

lst.append(10)
lst.append(20)
lst.append(30)
lst.insert(1, 15)

lst.display()        # [10, 15, 20, 30]

lst.remove(20)
lst.display()        # [10, 15, 30]

print(lst.pop())     # 30
lst.display()        # [10, 15]

print(lst.get(1))    # 15
lst.set(1, 100)
lst.display()        # [10, 100]
```

Time Complexity (important for exams)

Operation	Complexity
append	O(1) amortized
insert	O(n)
remove	O(n)
pop	O(n)
get/set	O(1)