1)

1. While loop runs len(lst) times and insert method runs len(lst) times.
   Therefore, len(lst) x len(lst) → $O(n^2)$

2. While loop runs n times and append method runs $\theta(1)$.
   Therefore, $O(n)$.

2.c (Ec)

3-b)

```python
def find_duplicates(lst):
    result = []
    biggest_num = max(lst)
    lst2 = (biggest_num + 1) * [0]

    for i in range(len(lst)):
        lst2[lst[i]] += 1

    for j in range(len(lst2)):
        if lst2[j] > 1:
            result.append(j)

    return result
```

In the worst case, max(lst) has O(n) and two for loop runs O(n).

Therefore, the worst case running time would be O(n).

O(n)

4-a) In the worst case, the removing and shifting runs n times and while loop runs n times.

$n \times n = n^2$, Therefore, the worst case running time is $O(n^2)$

4-c)

```python
def remove_all(lst, value=None):
    if value == None:
        raise ValueError

    count = 0
    for i in range(len(lst)):
        if lst[i] == value:
            count += 1
        elif lst[i] != value:
            lst[i-count] = lst[i]

    for j in range(count):
        lst.pop()

    return lst
```

In the worst case, the first for loop runs n times and the second for loop runs n times.

O(n)