Question 1

1. $1 + 2 + 3 + \cdots + n = \frac{n(n+1)}{2} = O(n^2)$

The while loop will run n times. Inserting would first cost 1, then the second time 2 since it has to shift 1 element and insert. The third time would cost 3 since it has to shift 2 elements and insert. This would go on until the while loop ends. Therefore the total cost would be $1 + 2 + 3 \cdots + n = \frac{n(n+1)}{2} = O(n^2)$

2. The while loop would run n times. Appending would add an element to the list until it has to resize the list. There fore the total cost would be $1 + 2 + 4 + 8 \ldots n = 2n - 1 = O(n)$.

Question 3

worst-case

The runtime would be O(n) since we would have to account for resizing due to the append method. The total cost would be $1 + 2 + 4 + 8 \ldots n = (2n-1) = O(n$

Question 4

a) The worst case runtime would be $O(n^2)$ ~~since~~ ~~time~~ If the input is a list containing all of the same elements, the while loop would run the length of the list and ~~they~~ will have to resize each time an occurance of the value is removed.

The total cost would be $n+(n-1)+(n-2)+\dots+1=\frac{n(n+1)}{2}$

$=O(n^2)$