Q1. 1. $T(n) = O(n^2)$

This is because for the while loop it's $T(n) = O(n)$ and for the insert operation it's $T(n) = O(n)$

$\therefore$ the total worst case run time is $T(n) = O(n \times n) = O(n^2)$

2. $T(n) = O(n)$

This is because for the while loop it's $T(n) = O(n)$ and for the append operation it's $T(n) = O(1)$

$\therefore$ The total worst case run time is $T(n) = O(n \times 1) = O(n)$

Q3. b) The worst case run time for my function would be O(n).

* This is because my function consists of 3 for loops that run parallel to each other with each for loop having $T(n) = O(n \times 1) = O(n)$ since the operation inside each loop is $O(1)$.

$\therefore$ the total worst case run time is $O(n+n+n) = O(3n) = O(n)$

Q4. a) The ~~worse~~ worst case run time is if we would have to remove all elements in the list and the loop would exit once the ValueError is raised. In this case we would have to iterate over the whole list. So $T(n) = O(n)$ for the outer while loop and $T(n) = O(n)$ for the inner remove function. Thus, the total run time in the worst ~~ca~~ case would be $T(n) = O(n \times n) = O(n^2)$.

c) The implementation ~~costs~~ consists of 2 parallel for loops with each having $T(n) = O(1)$ inside

* them for the operations performed. So for each loops it's $T(n) = O(n \times 1) = O(n)$.

The total worst case run time would be $T(n) = O(n+n) = O(2n) = O(n)$