Exercise 1

LI, Pengda
10225101460

1. Revise Landau's O and Θ notation. Explain why the bubblesort algorithm for sorting a list of length n is $O(n^2)$. Is it $\Theta(n^2)$?

The bubblesort algorithm can be described as follows:

Algorithm 1 Bubble Sort

```
Require: Array A, length n
for i = 1 to n − 1 do
    for j = 0 to n − i − 1 do
        if A[j] > A[j + 1] then
            Swap A[j] and A[j + 1]
        end if
    end for
end for
return Array A
```

The outer loop runs $n-1$ times and the inner loop runs $n-i-1$ times, where i is the index of the outer loop. The total number of comparisons is

$$(n-1) + (n-2) + \cdots + 1 = \frac{n(n-1)}{2}$$

Let $f(n) = \frac{n(n-1)}{2}$, then for any $n > 1$, we have

$$f(n) \leq n^2$$

Therefore, by the definition of O notation, the bubblesort algorithm is $O(n^2)$.

The bubblesort algorithm is $\Theta(n^2)$. To prove this, we need to show that the algorithm is both $O(n^2)$ and $\Omega(n^2)$.

Let $f(n) = \frac{n(n-1)}{2}$, then for any $n \geq 3$, we have

$$f(n) = \frac{1}{3}n^2 = \frac{n^2}{6} = \frac{n}{2} = \frac{1}{6}n(n-3) \geq 0$$

That is, $f(n) \geq Cn^2$ for $n \geq 3$ and $C = 1/3$. Therefore, the bubblesort algorithm is $\Omega(n^2)$.

So the bubblesort algorithm is $\Theta(n^2)$.

2. Prove that linear search for an element in an unordered list of length n is $\Theta(n)$.

The linear search algorithm can be described as follows:

Algorithm 2 Linear Search

Require: Array A, length n, element x

```
for i = 0 to n − 1 do
    if A[i] = x then
        return i
    end if
end for
return −1
```

In the worst case, the element x is at the end of the list or not in the list. The loop runs n times. Therefore, the linear search algorithm is $\Omega(n)$.

In the average case, the element x is in the list and the probability of finding it is $1/n$. The expected number of comparisons is $n/2$. Therefore, the linear search algorithm is $O(n)$.

Therefore, the linear search algorithm is $\Theta(n)$.