Part 1:
Time complexity

let from street and avenue, one of them is maximum so let it be n.

So, reoccurrence relation is when n is zero its takes constant time complexity.
otherwise, its T(n-1) + T(n-1) + 1

T(n) = 2T(n-1) + 1
T(n) = 2(2T(n-2) + 1) + 1 = 2^2T(n-2) + 2 + 1
T(n) = 2(2(T(n-3) + 1) + 1) + 1 = 2^3T(n-3) + 2^2 + 2 + 1
.....
T(n) = 2^(n-1)T(1) + 2^(n-2) + ... + 2^2 + 2 + 1
T(n) = 2^nT(0) + 2^(n-1) + ... + 2^2 + 2 + 1

T(0) = 1
So,

T(n) = 2^n(1) + 2^(n-1) + ... + 2^2 + 2 + 1
T(n) = 2^n + (2^(n-1) -1)/2

so, we can write it as T(n) = 2^n
So, the time complexity for this function is 2^n

Part 2:
Time complexity

Here, the time complexity for this function depends on the length of T most of the time.
So, when T or S is empty, complexity will be constant and otherwise
let n is the total length of T, so from T(n) the reoccurrence relation should be
T(n) = T(n-1) + 1   (because complexity depends on size of T)
where n is size of array and the n goes from size to first element of array, so

T(n) = T(n-2) + 1 + 1
T(n) = T(n-3) + 1 + 1 + 1
....
T(n) = T(1) + 1 + 1 + ... + 1
T(n) = T(0) + 1 + 1 + 1 + ... + 1

so, T(n) = n(1)

The time complexity of this function is O(n).