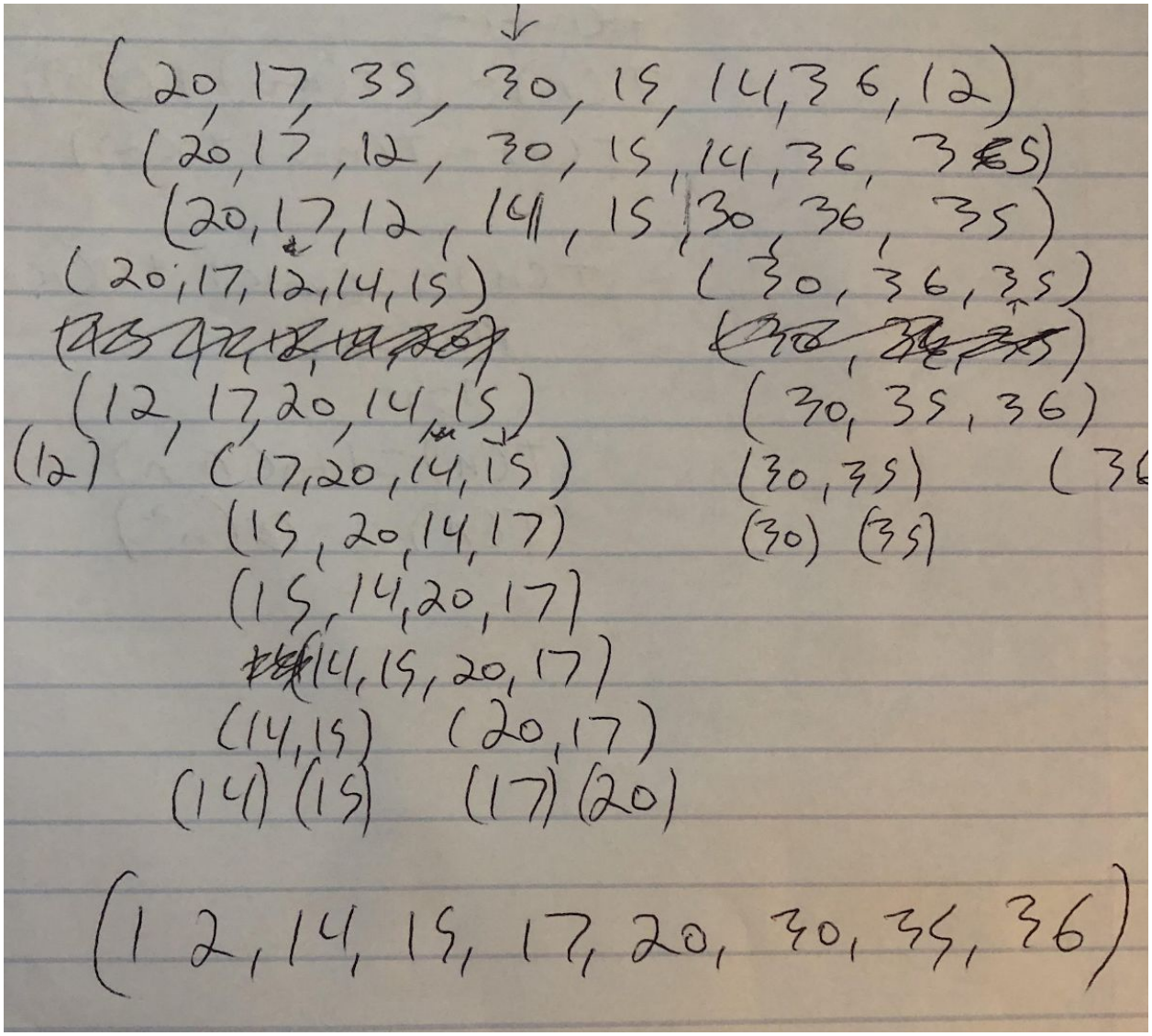
1.



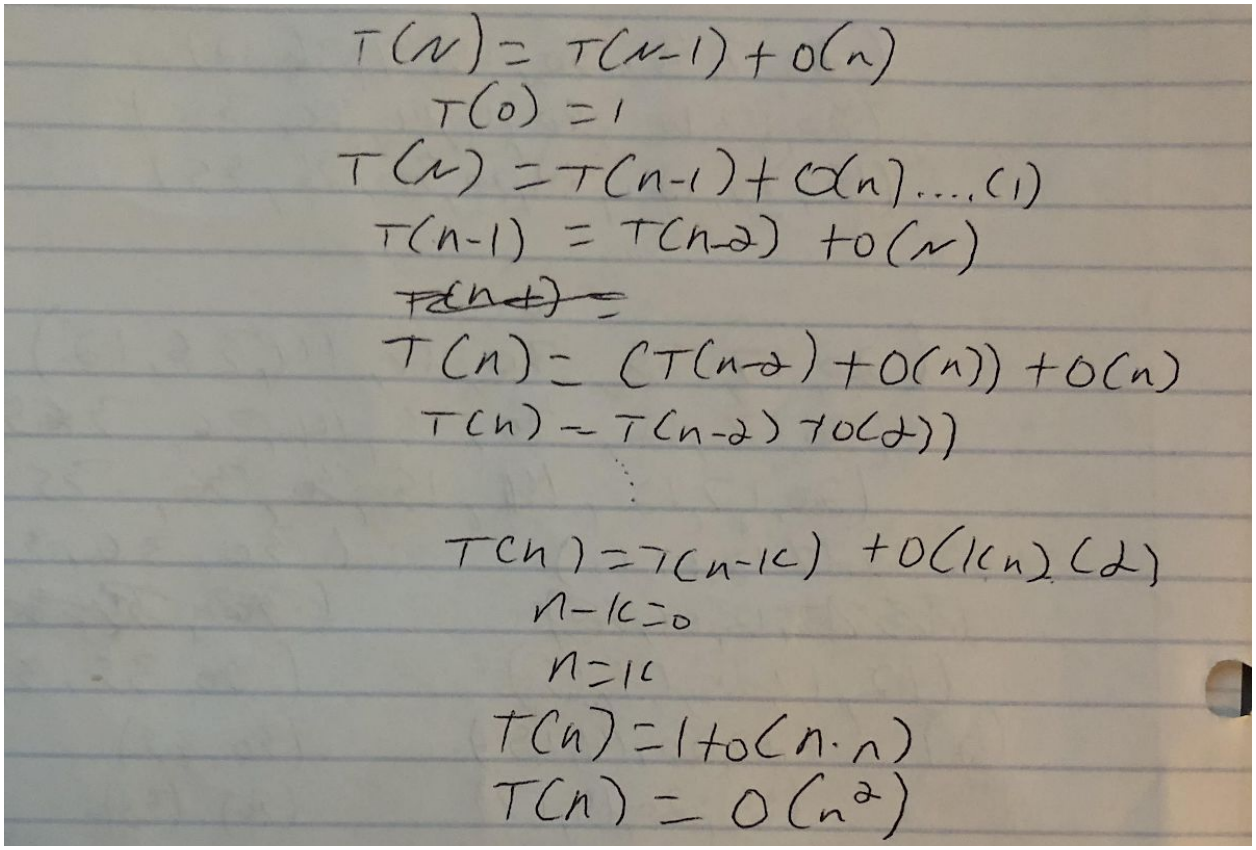
2.

```
def recursive_insertion_sort(array,length):
    if length<=1:
        return
    recursive_insertion_sort(array,length-1)
    last_element = array[length-1]
    x = length-2
    while (x>=0 and array[x]>last_element):
        array[x+1] = array[x]
        x = x-1
    array[x+1]=last_element
```

T(n) = T(n-1)+O(n)

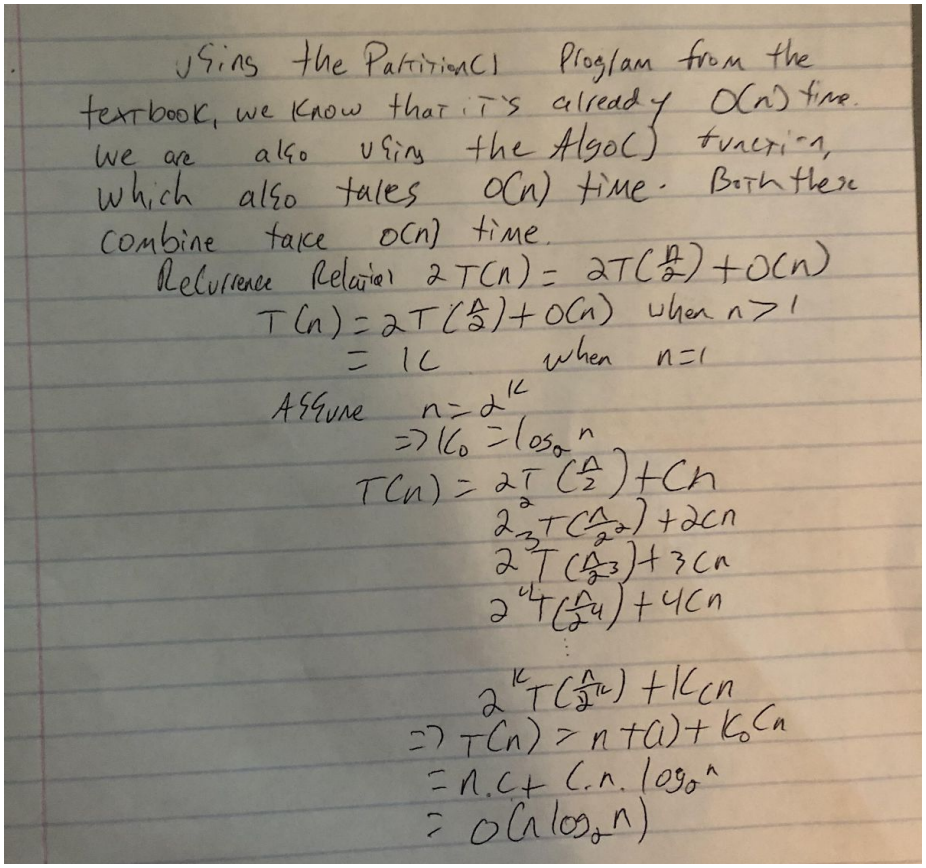
T(0)=1

T(n) = T(n-1)+O(n) ... (1)



3.

```
modified_quick_sort(array,low,high){
    if(low<high){
        pivot = Algo(array)
        Swap(array,pivot,last) //Swap function from textbook
        parted = partition(array,low,high,pivot) //partition function from textbook
        modified_quick_sort(array,low,parted-1)
        modified_quick_sort(array,parted + 1,high)
    }
}
```



4.

1. Find middle element of array
2. If this middle element is equal to i, then done
3. If not,
   a. then do the same for the subarray on the left
   b. Do the same for the sub array on the right
5. If after going through all of them, none are found, then it does not exist.