**GROUP MEMBERS**

1. JESSE KANYI- SCT221-0246/2022
2. MAX MURAGE- SCT221-0212/2022
3. TITUS GICHEHA- SCT221-0216/2022

**QUESTION**

**Design an algorithm to merge two sorted lists that are passed as parameters, and return one merged sorted list.**

function merge\_sorted\_lists(list\_one, list\_two):

merged\_list = []

i = 0

j = 0

while i < len(list\_one) and j < len(list\_two):

if list\_one[i] <= list\_two[j]:

merged\_list.append(list\_one[i])

i += 1

else:

merged\_list.append(list\_two[j])

j += 1

while i < len(list\_one):

merged\_list.append(list\_one[i])

i += 1

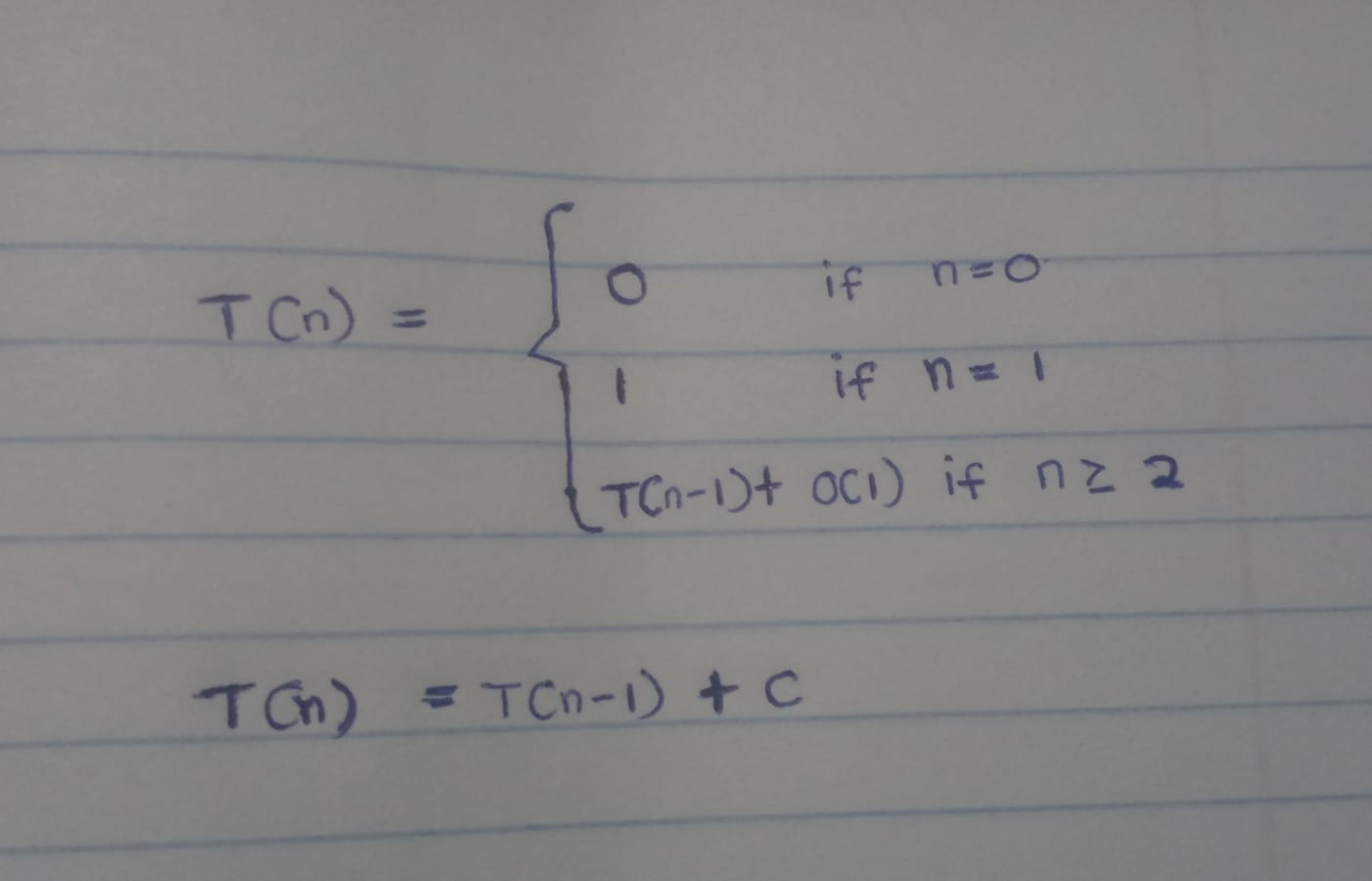
while j < len(list\_two):

merged\_list.append(list\_two[j])

j += 1

return merged\_list

**Derive a recurrence relation for this algorithm.**



**Estimate time and space complexities for the same algorithm.**

**Time complexity**

O(n) as all elements are compared and appended exactly once.

**Space complexity**

The primary data structure used is the merged\_list, which stores the combined elements of the input lists. The merged\_list will contain all n elements. Hence, the space complexity is **O(n)**.