**Assignment**

1. **Write an efficient recursive algorithm that takes a sentence , starting index and ending index. The algorithm should then return a sentence that contain words between the starting and ending indices. Write recurrence relation of your algorithm and find time complexity using tracing tree method**

**Answer**

Lets define (T(n)) as the time complexity of the algorithm, where (n) is the number of words between the start and end indices.

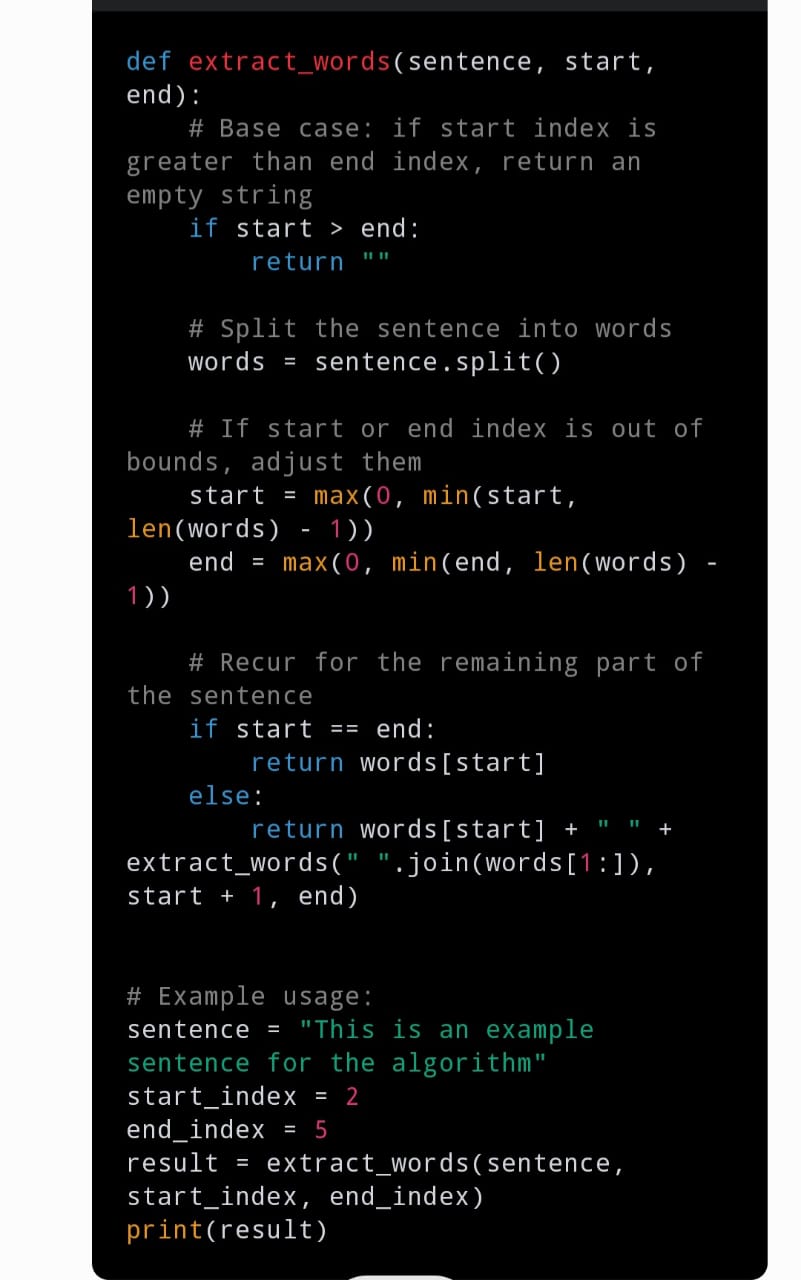
The Algorithm splits the recurrence into words at each recursion where (n) is the number of words between the start and end indices.

The algorithm splits the recurrence into words at each recursion, reducing the length of the sentence by (1) in each recursive call until the base case is reached. Therefore, the recurrence relation can be expressed as

T(n) = T(n-1) + 0(n)

Here, (0(n)) represents the time taken to split the sentence into words, which is linear in the number of words.

Here is an efficient recursive algorithm in python that takesa sentence, a starting index, and ending index. The algorithm returns a sentence containg words between atarting and ending indices



**TIME COMPLEXITY ANALYSIS USING TRACING TREE METHOD**

Lets trace the recursive calls to find the time complexity.

(T(n) = T(n-1) + 0(n)

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

T/dots)

(T(2) = T(1) + 0 (2))

Now lets substitute these back into the original equation

T(n) = T(n-1) + 0(n)

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

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

Now, we sum the terms;

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

This simplifies to ;

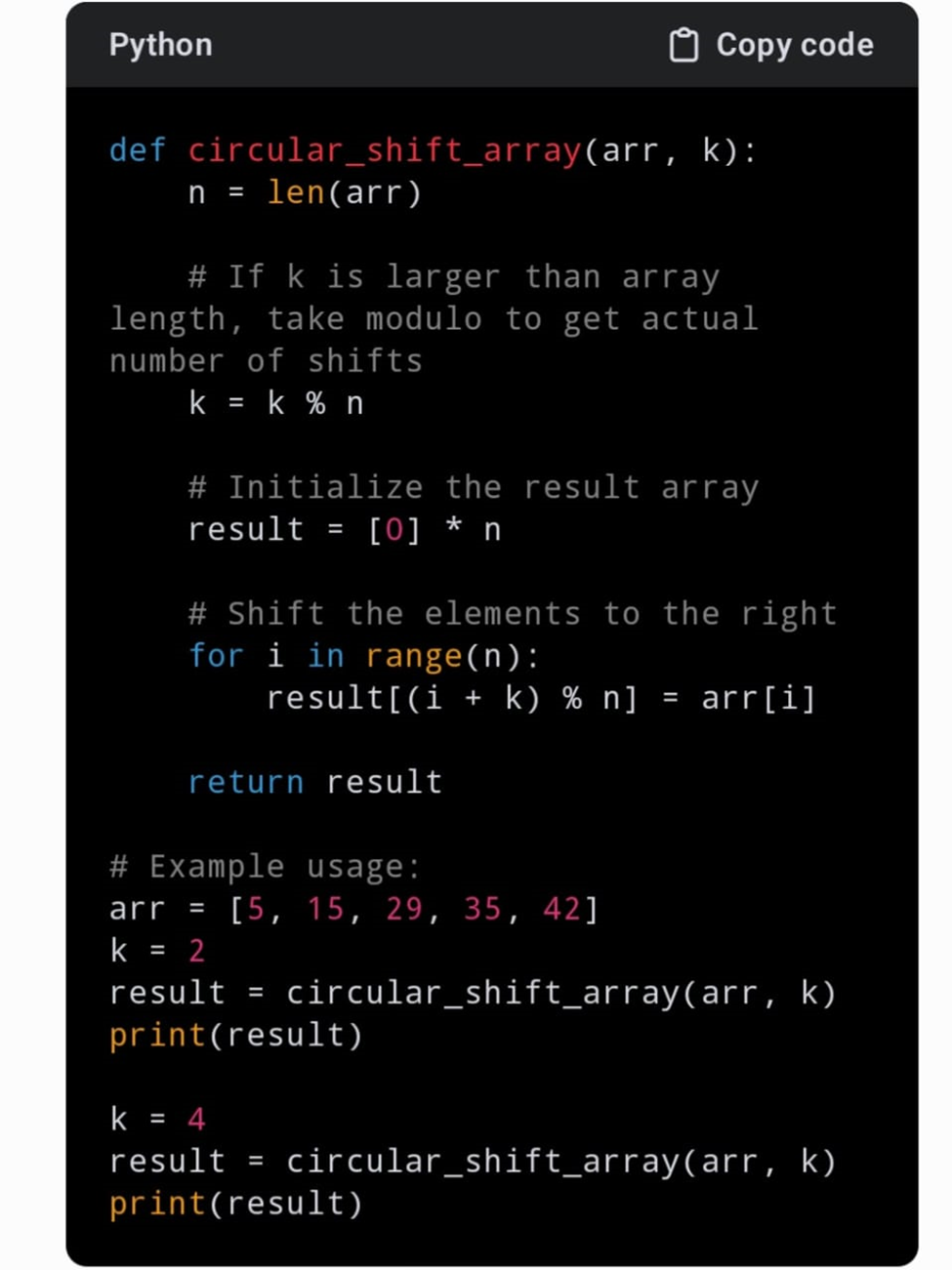
T(n) = 0(n^2)

So the time complexity of the algorithm is (0(n^2)) when analysed using the tracing tree method.

1. **Write an efficient algorithm that takes an arrays A(n)…..n) of sorted integers and return an array with elements that have seen circurlaly shifted k positions to the night for example a sorted array A= [ 5, 15, 29, 35, 42,) is converted to A [35,42,5,12,27,29] after circular shifted 2 positions, while the same array A = [5, 15, 29,35,42] is converted to A [ 27,29,35,42,5,15,] after circurlaly shifted 4 positons, write the recurrence relation of your solution and find time complexity of your algorithm using iterative method.**

**Answer**

Here is an efficient algorithm in python that takes a sorted arrays of integers and the number of positions to circurlaly shift the array to the right

****

**Recurrence relation**

Lets define T(n) as the time complexity of the algorithm, where (n) is the number of elements in the input array.

The Algorithm loops through each element of the input array once to perform the circular shift operation. Therefore the recurrence relation can be expressed as

T(n) = 0(n)

Here (0(n)) represents the time taken to perform the circular shift operation.

**Time complexity analysis using iterative method**

The time complexity of the algorithm is linear because it loops through each element of the input array once.

Therefore, the time complexity is (0(n)), where n is the number of elements in the input array once.

Therefore, the time complexity is (0(n), where n is the number of elements in the input array. This is because the algorithm takes linear time to perform the circular shift operation regardless of the value of k