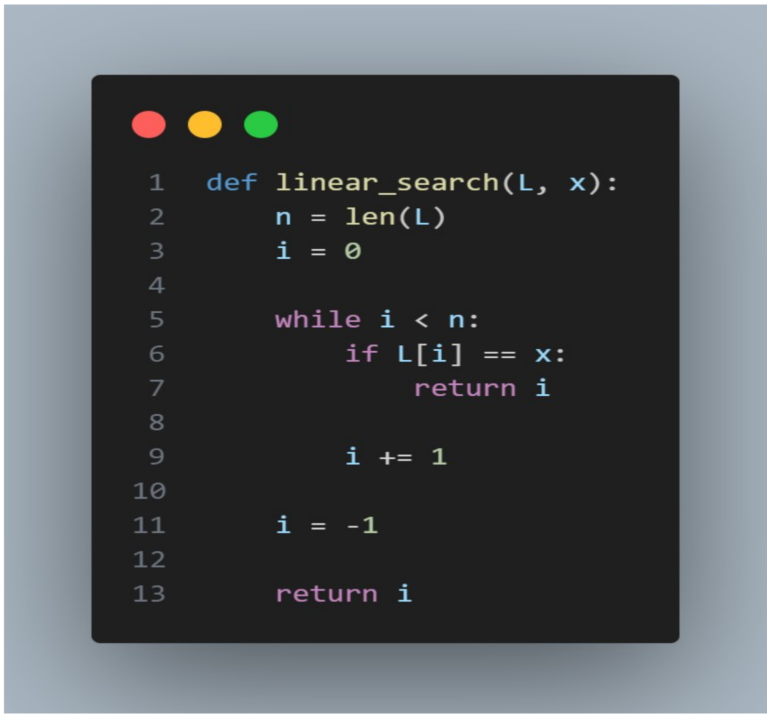
Linear Search Algorithm



```python
def linear_search(L, x):
    n = len(L)
    i = 0

    while i < n:
        if L[i] == x:
            return i

        i += 1

    i = -1

    return i
```

The Linear Search algorithm is one of the simplest techniques used to find a specific element (called the "target") within a list or array. It works by sequentially checking each element of the list until the desired element is found or the end of the list is reached.

Algorithm Steps

1. Start at the first element of the list.
2. Compare the target element with the current element.
3. If the current element matches the target, return its position (index).
4. If the current element does not match, move to the next element.
5. Repeat steps 2–4 until the target is found or the list ends.
6. If the target is not found, return a failure result (e.g., `-1`).

Key Characteristics

- **Time Complexity**:
  - Best Case: O(1) (when the target is the first element).
  - Worst Case: O(n) (when the target is the last element or not present).
- **Space Complexity**: O(1) (no extra memory is required).
- Works on both **sorted** and **unsorted** lists.

Advantages

- Easy to implement.
- No need for preprocessing the list (e.g., sorting).

Disadvantages

- Inefficient for large lists compared to more advanced algorithms like binary search.