The linear search algorithm, also known as sequential search, is a simple searching algorithm that checks each element in a collection until the target element is found or the entire collection has been traversed.

Here's how the linear search algorithm works:

1. Start at the beginning of the collection.
2. Compare the target element with the current element in the collection.
3. If the current element matches the target element, the search is successful. Return the index or position of the element.
4. If the current element does not match the target element, move to the next element in the collection.
5. Repeat steps 2-4 until either the target element is found or the end of the collection is reached.

Here's the implementation of the linear search algorithm in Python:

```
def linear_search(collection, target):
    for i in range(len(collection)):
        if collection[i] == target:
            return i  # Return the index if the target is found
    return -1  # Return -1 if the target is not found

# Example usage
data = [5, 3, 8, 4, 2, 7, 1, 6]
target = 4
index = linear_search(data, target)
if index != -1:
    print(f"Target {target} found at index {index}")
else:
```

```
    print(f"Target {target} not found in the collection")
```

In this example, the linear_search function takes a collection (e.g., a list) and a target element as input. It iterates through each element in the collection using a for loop and checks if the current element matches the target. If a match is found, it returns the index of the element. If the loop completes without finding a match, it returns -1 to indicate that the target element was not found.

The linear search algorithm has a time complexity of O(n), where n is the size of the collection. In the worst case, it may need to traverse the entire collection to find the target element.