

Exercise 1

- An algorithm is a set of clearly defined instructions or step-by-step processes designed to perform a specific task or solve a specific problem.
- Algorithms relate to problem-solving:
 - **Structured Approach:** Algorithms provide a clear, structured way to tackle problems, breaking them down into smaller, manageable steps.
 - **Efficiency:** They help find the most efficient solution, optimizing time and resources.
 - **Reproducibility:** Algorithms ensure consistent results, enabling the same problem to be solved repeatedly with the same steps.
 - **Automation:** They allow problems to be solved automatically by computers, enhancing speed and accuracy.
 - **Scalability:** Algorithms can handle large-scale problems by applying the same principles to larger datasets or more complex scenarios.
 - **Innovation:** Developing new algorithms can lead to innovative solutions and advancements in various fields.

Exercise 2

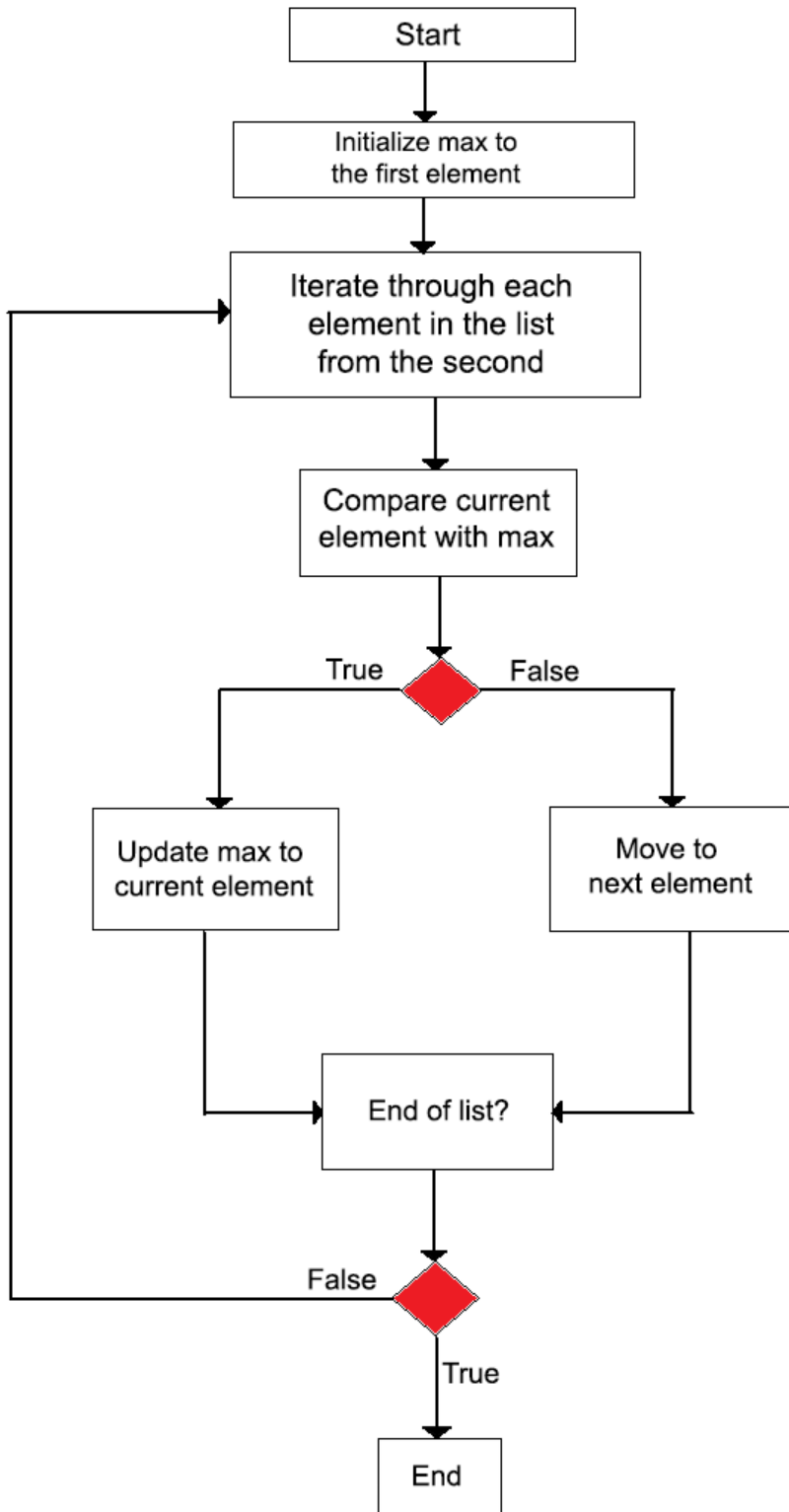
- **Sequence:**
 - **Definition:** A sequence is a set of instructions that are executed in order, one after another.
 - **Use in Algorithms:** It represents the linear flow of control, where each step follows the previous one without any branching.
- **Selection:**
 - **Definition:** Selection involves making a decision and executing a certain part of the code based on a condition
 - **Use in Algorithms:** It allows for branching, where different paths can be taken depending on the evaluation of conditions.
- **Repetition:**
 - **Definition:** Repetition involves executing a set of instructions repeatedly, either a specified number of times or until a condition is met (often using loops like for or while).

- **Use in Algorithms:** It enables repeated execution of code blocks, which is essential for tasks that require iteration.

Exercise 3

Description of Diagram:

- **Start:** Algorithm starts.
- **Initialize max:** Set max to the first element.
- **Iterate through the list:** Begin iteration from the second element.
- **Compare the current element with max:** Check if the current element is greater than max.
- **Update max:** If the condition is true, update max.
- **Move to the next element:** Proceed to the next element in the list.
- **End of List?:** Check if the end of the list is reached.
- **End:** If the list has been fully traversed, the algorithm ends with max holding the largest number.



Exercise 4

Explanation

- Read the first integer: This step prompts the user to enter the first integer, stored in variable a.
- Read the second integer: Similar to the first step, the user is prompted to enter the second integer, which is stored in variable b.
- Calculate the sum of a and b: The core operation, where the values of a and b are added together and the result is assigned to the variable sum.
- Output the result: Finally, the algorithm displays a message indicating the sum of the two integers that the user inputs.

```
Algorithm SumOfTwoIntegers
Input: Two integers, a and b
Output: The sum of a and b

Begin
    // Step 1: Read the first integer
    Read a

    // Step 2: Read the second integer
    Read b

    // Step 3: Calculate the sum of a and b
    sum = a + b

    // Step 4: Output the result
    Print "The sum of", a, "and", b, "is", sum
End
```

Exercise 5

```

Algorithm SumOfList
Input: A list of integers, list
Output: The sum of all integers in the list

Begin
    // Step 1: Initialize sum to 0
    sum = 0

    // Step 2: Iterate through each element in the list
    For each element in list do
        // Step 3: Add the current element to sum
        sum = sum + element
    End For

    // Step 4: Output the result
    Print "The sum of the list is", sum
End

```

Explanation:

- **Initialize Sum:** The variable sum is initialized to 0.
- **Iterate through the list:** A loop goes through each element in the list.
- **Add each element to Sum:** Within the loop, each element is added to sum.
- **Output the result:** After the loop ends, the final value of sum is printed.
- **End:** The algorithm terminates, having calculated and displayed the sum of the list.

Exercise 6

Pseudocode

```
Algorithm FindLargest
Input: A list of integers, list
Output: The largest integer in the list

Begin
    // Step 1: Initialize largest to the first element of the list
    largest = list[0]

    // Step 2: Iterate through each element in the list starting from the second element
    For i = 1 to length(list) - 1 do
        // Step 3: Compare current element with largest
        If list[i] > largest then
            // Update largest to the current element
            largest = list[i]
        End If
    End For

    // Step 4: Output the result
    Print "The largest number in the list is", largest
End
```

UML diagram

