Algorithm Explanation for JSON Data Table and Chart

1. Function: loadJSON

This function loads a JSON file using the Fetch API.

1. fetch('data.json'): Makes an HTTP request to fetch the JSON file from the specified path.
2. .then(response => response.json()): Once the file is loaded, the response is converted to JSON.
3. .then(data => { processJSON(data) }): The parsed JSON data is passed to another function for processing.
4. .catch(error => console.error()): If an error occurs, it is logged in the console.

2. Function: processJSON

This function processes the JSON data and updates the table and chart.

1. const tableBody = document.querySelector('#dataTable tbody'): Selects the table body to update it dynamically.
2. tableBody.innerHTML = '': Clears any previous content from the table.
3. data.categories.forEach(item => {...}): Iterates over the 'categories' array inside the JSON object.
4. Creates new table rows and cells for each category and value, then appends them to the table.
5. chartData.push(...): Collects data for the chart in an array for later use.
6. drawSimpleChart(chartData): Calls the function to draw the chart after the table is rendered.

3. Function: drawSimpleChart (Detailed)

This function draws a simple bar chart using HTML5 canvas. Below is a detailed explanation using the same values from your code.

1. const canvas = document.getElementById('simpleChart'): This selects the canvas element from the HTML file, where the chart will be drawn.

2. const ctx = canvas.getContext('2d'): This retrieves the 2D context of the canvas, allowing us to draw shapes like bars.

3. ctx.clearRect(0, 0, canvas.width, canvas.height): Clears any existing drawing on the canvas to prevent overlap. It's like resetting the canvas.

4. const maxValue = Math.max(...data.map(item => item.value)): Finds the highest value in the dataset to ensure bars are scaled proportionally. In this case, the maximum value is 60 (from 'Grapes').

5. Iterating over data (forEach loop):

   - For each item (e.g., 'Apples', 'Bananas'), the bar height is calculated as:
   barHeight = (item.value / maxValue) * maxBarHeight
   - For 'Apples' with value 50, the bar height is: (50 / 60) * 250 = ~208.33px.
   - For 'Bananas' with value 30, the bar height is: (30 / 60) * 250 = 125px.

6. ctx.fillRect(x, y, barWidth, barHeight): This draws a rectangle for each item at the calculated x and y positions, with a width of 50px and the calculated height. The 'x' position ensures bars don't overlap and are evenly spaced.

7. ctx.fillText(...): Labels are added for each category ('Apples', 'Bananas') and their values (50, 30) at the appropriate positions on the canvas, making the chart easier to read.