Specialisation Assessment Questions

Redux [25 Marks]

Part 1 [5 Marks]

1. **Can you provide a brief summary of what is happening in this function code?**

    a. This function returns a new state object with a value that's been increased by one, so long as the action type is increment.

2. **Add one action that tells the reducer to reduce the state value by 1**

```
function countReducer(state = initialState, action) {
  if (action.type === "increment") {
    return {
      value: state.value + 1
    };
  } else if (action.type === "decrease") {
    return {
      value: state.value - 1
    };
  }
  return state;
}
```

3. **Add one action that tells the reducer to reset the state**

```
function countReducer(state = initialState, action) {
  if (action.type === "increment") {
    return {
      value: state.value + 1
    };
  } else if (action.type === "decrease") {
    return {
      value: state.value - 1
    };
  } else if (action.type === "reset") {
    return {
      value: initialState.value
    };
  }

  return state;
}
```

Part 2 [10 Marks]

This section handling state locally.

```
const classInfo = () => {
    let [studentsCount, setStudentsCount] = useState(0);
    return(
        <div>
            <p>Number of students in class room: {studentsCount
            <button onClick={?????} >Add Student</button>
        </div>
    )
}
```

In the code above the useState hook is used to set the state of a variable inside the

component.

1. **Can you provide a brief summary on what is happening on line 34, 39?**
    a. On line 34, he code uses the useState hook to declare the state variable called studentsCount. studentsCount state is updated by setStudentsCount

   b. On line 39, there is a button with an onClick attribute, however the function that is supposed to be completed on click is just a placeholder ‘????’. The button currently won’t do anything

2. **When a user clicks on the “Add student” button update the state (studentsCount) to include only the total number of students who are present. Using the data provided below:**

   1. **Write a pseudocode of how your function would look.**

      ```
      const classInfo = () => {
        let [studentsCount, setStudentsCount] = useState(0);

      // Function to increment student count
        const addStudent = () => {
          const updatedCount = studentsCount + 1;
              // Updates state studentsCount
          setStudentsCount(updatedCount);
        };

        return (
          <div>
            <p>Number of students in class room: {studentsCount}
            <button onClick={addStudent}>Add Student</button>
          </div>
        );
      };
      ```

   2. **How do you ensure that the function is triggered when the button is clicked?**

      a. To ensure the function is triggered, when the button is clicked, the ???? placeholder needs to be removed and the addStudent function should be added to the onClick event

   3. **How will you update the state with the result of your function?**

    a. Use the setStudents function to update the state by calling it with updatedCount

> 💡 Note to Marker: I misunderstood the question and this answer updates the studentCount state on the click of a button rather than only including the number of students present. I don’t have time to change it

Part 3 [10 Marks]

Now let’s use dispatch to update the state on button click

```
const initialState = { value: 0 }
function countReducer(state = initialState, action){
        if (action.type === 'increment'){
            return{
                value:state.value + action.payload
            }
        }
    return state
}
```

1. **A change of code was made on line 174 (figure 4), can you briefly explain what that would do?**

    a. Instead of incrementing by 1, it adds the value of action.payload to the state’s value but again only if the action type is increment. Now when you dispatch an action with type 'increment', you also have to include a payload. For example, if you dispatch an action like { type:'increment', payload : 6 } it will increment the value by 6.

2. **Let’s say we don’t want to set the state locally anymore and want to use dispatch. How would you ensure that an “increment” action that also contains the result of the studentCount is dispatched on button click? According to your answer in part 2.2b what would need to be changed?**

3. **Which code do you think is best suited to ensure that the "increment" action updates the state with the correct total number of students who are present. Is it Figure 4? Or Figure 5? Explain the code difference and your reasoning**

Algorithms 1 (Coding) [15 Marks]

Write an algorithm that returns true if the given string is a palindrome. Otherwise, return false.

Note: A String is said to be a palindrome if the string is spelled the same way forward and backwards. For example, some sample input and outputs would be:

	stringA value	Output value
Sample Input 1	radar	True
Sample Input 2	level	True
Sample Input 3	Pencil	False
Sample Input 4	a	True

```
function isPalindrome(string) {
    // Convert the string to lowercase to make the comparison ca
    const lowercaseString = string.toLowerCase();

    // Split string into array, reverse order and then rejoin.
    const reversedString = lowercaseString.split('').reverse().j
    // Compare if reversed string is same as formatted input str
    return lowercaseString === reversedString;
  }

  // Test cases
  console.log(isPalindrome("radar"));   // TRUE
  console.log(isPalindrome("level"));   // TRUE
  console.log(isPalindrome("Pencil"));  // FALSE
  console.log(isPalindrome("a"));       // TRUE
    console.log(isPalindrome("abcdcba")) // TRUE
    console.log(isPalindrome("aba")) // TRUE
    console.log(isPalindrome("c")) // TRUE
    console.log(isPalindrome("radar")) // TRUE
    console.log(isPalindrome("level")) // TRUE
    console.log(isPalindrome("pencil")) // FALSE
    console.log(isPalindrome("ark")) // FALSE
    console.log(isPalindrome("aa")) // TRUE
```

In your answer, please discuss your solution - what is its Big O Time & Space complexity? Why have you chosen this approach? Could there be a more efficient way (and if so, how)?

1. Converting the string to lowercase takes O(n) time
2. Splitting the string into an array takes O(n) time
3. Reversing the array takes O(n) time
4. Joining the array back into a string takes O(n) time

In terms of Time Complexity, these are all O(n) because n is the length of the input string and the time it would take to complete all of these tasks is dependent on the length of the string

1. The lowercaseString variable takes O(n) space
2. The reversedString variable also takes O(n) space

In terms of Space Complexity, both of these variables are O(n) where n is the length of the input string. They are both O(n) as they store a copy of the input string, just in different orders. The same amount of space is take by both variables

The overall Time and Space Complexity is O(n). This is quite efficient. I don't believe there is a more efficient way to approach this algorithm. Possibly comparing the characters in the string from both ends instead of creating a reversed copy, however this wouldn't change the overall Time Complexity

Algorithms 2 (Coding) [20 Marks]

Write a function that takes in an unsorted array of any size. These elements are in the range of 1 to n. In the input array one number is missing. Your function should return the missing number.

If the input array contains a negative number or non-numeric value then return an error with the correct error message.

For example, some sample input and outputs would be:

	Array input	Output
Sample Input 1	[4,5,1,3, 5]	Missing = 2
Sample Input 2	[4, 3,5, 6, 8, 2, 1, 3]	Missing = 7
Sample Input 3	[1,2,3,4]	Noting is missing
Sample Input 4	[4,5, -1,3, 5]	Invalid input, negative number detected
Sample Input 5	[3, 4, 5, 6, 'cfg']	Invalid input, non-numeric value detected

```
function isMissing(inputArray) {
  // Loop to check for non-numeric values or negative numbers
  for (let i = 0; i < inputArray.length; i++) {
    if (typeof inputArray[i] !== "number" || inputArray[i] < 0)
      return "Invalid input, non-numeric value or negative numbe
    }
  }
  // Sort the array in ascending order by comparing a and b elem
  let sortedArray = inputArray.sort((a, b) => a - b);
  console.log(sortedArray)

  // Remove duplicates from array using Set
  let uniqueSortedArray = [...new Set(sortedArray)]
  console.log(uniqueSortedArray)

  // Check for missing number in sorted array
  for (let i = 0; i < uniqueSortedArray.length; i++) {
    if (uniqueSortedArray[i] !== i + 1) {
      return "Missing = " + (i + 1);
    }
  }

  // If no missing number is found, return a message
  return "Nothing is missing";
}
// Test cases
console.log(isMissing([4, 5, 1, 3, 5]));    // Missing = 2
console.log(isMissing([4, 3, 5, 6, 8, 2, 1, 3]));    // Missing
console.log(isMissing([1, 2, 3, 4]));    // Nothing is missing
console.log(isMissing([4, 5, -1, 3, 5]));    // Invalid input,
console.log(isMissing([3, 4, 5, 6, 'cfg']));    // Invalid input
```

In your answer, please discuss your solution - what is its Big O Time & Space complexity? Why have you chosen this approach? Could there be a more efficient way (and if so, how)?

1. Checking for non-numeric values or negative numbers takes O(n) time
2. Sorting the array takes O(n log n) time
3. Removing duplicates using Set takes O(n) time
4. Checking for a missing number in sorted array takes O(n) time

In terms of Time Complexity, 1, 3 and 4 are all O(n) because n is the length of the elements in the array and the time it would take to complete all of these tasks is dependent on the length of the array. However sorting the array is O(n log n) because it compares each element one by one and is also dependent on the length of the array. This is very inefficient

1. The sortedArray variable takes O(n) space
2. The uniqueSortedArray variable takes O(n) space

The overall Time and Space Complexity is O(n log n) which is very inefficient. This is due to the Array sorting method. To reduce this, you would need to find a way to check if a number is missing without putting the array in any kind of order, possibly using maths instead of sorting however I don’t know how I would go about this.