

**FULLSTACKASSESSMENT-2HOURS**

|  |  |  |
| --- | --- | --- |
| **SECTION TYPE** | **TOTAL MARKS AVAILABLE** | **NOTES** |
| **Redux (pseudocode code/ reasoning explanation)** | 25 | Multiple questions, all comprising 25 total |
| **Algorithms 1 (Coding)** | 15 | 1 question only |
| **Algorithms 2 (Coding)** | 20 | 1 question only |
|  | **60 marks available total** |  |

*Questions begin on the next page.*

|  |  |
| --- | --- |
| **Redux** | **25 MARKS** |

This question involves using built-in React Hook for a simple attendance app. *Here are notes to help: -*

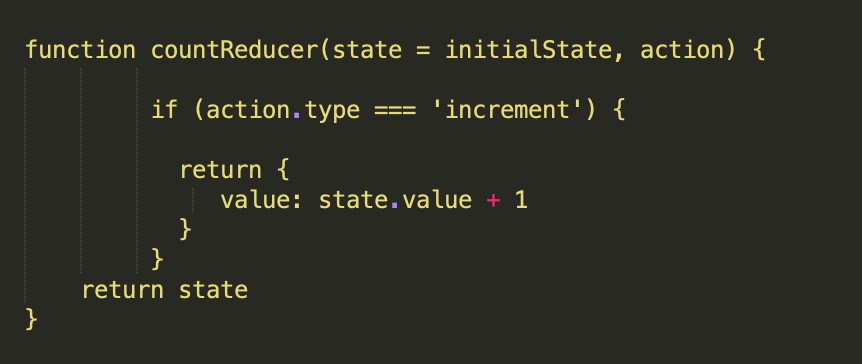
* *useState() : This allows you to have states variables in functional components. It helps to set and retrieve the state.*
* *A reducer: This is a function that returns some state data, triggered by an action type.*
* *An action: This is dispatched by components and is represented as one object that contains type property and sometimes payload property. It tells the reducer how to change the state. Here is an example of the shape of an action -> { type: 'GREETINGS', payload: ‘Hello’}*
* *Dispatch: this helps update the state by sending the type of action to the reducer function for it to perform its job. To invoke a dispatch function, you need to pass action as an argument to the dispatch function, e.g. dispatch ({type: "SOMETHING", payload: “SOMETHING” })}*

*\*\* Remember: Submit pseudocode or simply describe the solution.*

**Part 1 (5 marks):**

*Figure*

*1*



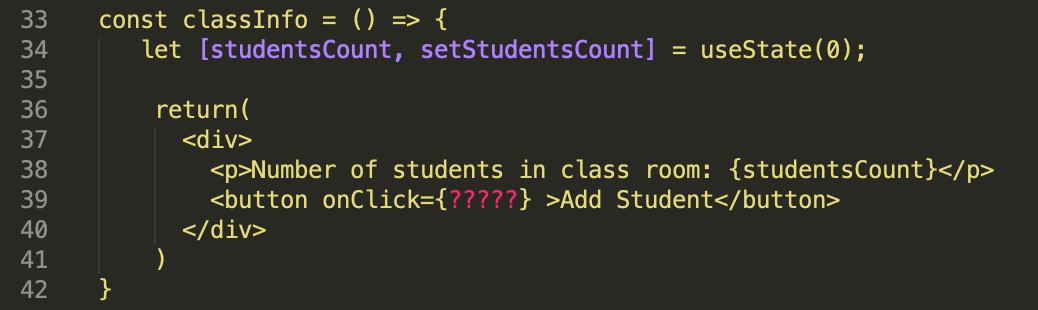
1. Can you provide a brief summary of what is happening in this function code?

This function is a reducer which takes the current state and action and returns a new state based on that action which in this instance is an ‘initialState’ – which will remember the changes made to the components state in the memory. It’s a count reducer so it is expecting a numerical value. The action type here is ‘increment’ which is defined in the return as a value of the state value +1 so it will return the original value in increments of +1. It

1. Add one action that tells the reducer to reduce the state value by 1
2. You would need to add after the return brackets “
3. else if (action.type === ‘decrement”)
4. return { value: state.value -1}
5. };
6. Return state;
7. Add one action that tells the reducer to reset the state
8. else if (action.type === ‘decrement”)
9. return { value: state.value -1}
10. };
11. Else if (action.type === ‘reset’) {
12. Retun initial.state; }
13. Return state;

**Part 2 (10 marks):**

This section involves handling state locally.



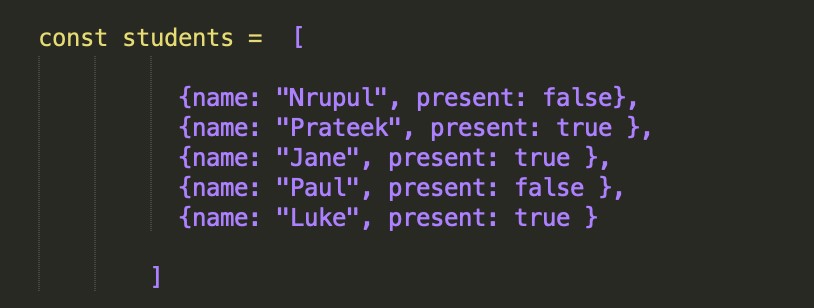
*In the code above the useState hook is used to set the state of a variable inside the component.*

1. Can you provide a brief summary on what is happening on line 34, 39?

The code is defining the ‘ClassInfo’ functional component as shown by ‘const’. Inside the component, there is a state variable ‘studentsCount’ intialised to 0 by the hook ‘useState(0)’. There is also a state updater function ‘setStudentCount’ as shown by the ‘set’ commonly used in redux.

The component returns a <div>, which contains a paragraph (<p>) that displays the current value of the StudentsCount, and then has a button inside it. The button has an event handler (on click) which I am guessing when the code was complete would add another student to the total number of the students in the room.

1. 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:



*Figure 3*

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

let [studentsCount, setStudentsCount] = useState(0);

const addStudent = () => {

setStudentCount( [students])

};

return(

<div>

<p>Number of students in class room: (studentsCount)</p>

<button onClick={addStudent} > Add Student </button>

</div>

)

}

* 1. How do you ensure that the function is triggered when the button is clicked?
  2. The function is called by the event handler OnClick – see pseudocode.
  3. How will you update the state with the result of your function?

Add the following code:

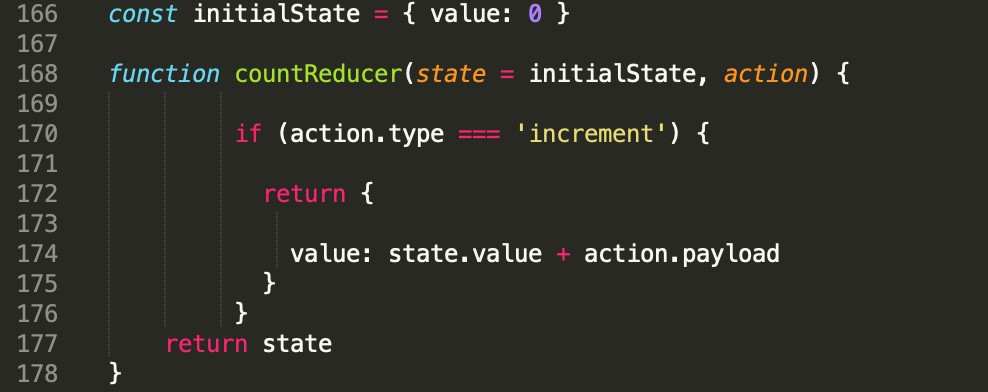
Else if (action.type === ‘reset’) {

Retun initial.state; }

Return state;

**Part 3 (10 marks):**

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



*Figure 4*

1. A change of code was made on line 174 (figure 4), can you briefly explain what that would do? The action.payload property has been added to the return value. This allows additional data to be stored and passed to the reducers so that any necessary more complicated actions can be dispatched. When dispatching the action (returning the value) with the payload data , it allows the function to access the data in the reducer; the action.payload function can hold any data type relevant to the action it could be a single object or an array in this example it would be the previous student data increments either using the save value +1 or the [student] data.
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? For a start Const = Dispath would have to be used.

let [studentsCount, setStudentsCount] = useState(0);

const addStudent = () => {

setStudentCount( [students])

};

return(

<div>

<p>Number of students in class room: (studentsCount)</p>

<button onClick={addStudent} > Add Student </button>

</div>

)

}

Const dispatch = Use Dispatch();

Let storedStudentCount = useSelector (state => state.StudentCount)

Let stored.addStudent = useSelector (state => state.addStudent)

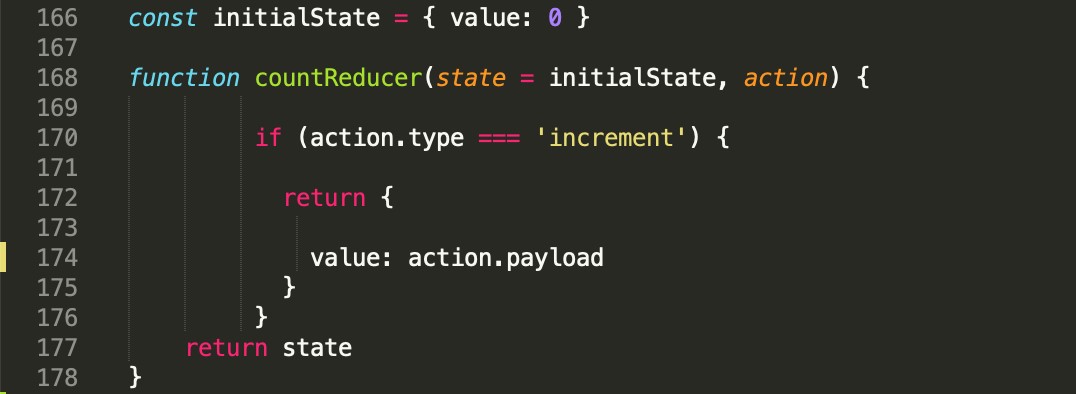
Const storedStudentCount = localstorage.getItem(‘StudentCount’)

Const stored.addStudent = localstorage.getItem(‘addStudent’)

Dispatch { type ‘SetStudentCount, Add Student’ })

1. 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*



Its figure 4. The difference between figure 4 and figure 5 is in figure 5 there is the absence of the state.value on line 174 “ value:action.payload”. in this setting the action.payload property has nothing to apply to because the state value is missing. If there is no state value, there is nothing for the payload to apply to in the reducer and the code will not run. When the code returns to state the button will not be updated.

*Figure 5*

|  |  |
| --- | --- |
| **Algorithms 1 (Coding)** | **15 MARKS**  **(1 question)** |

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 |

*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 e cient way (and if so, how)?*

*If you are short on time, you can also submit pseudocode or simply describe what solution you’d write in code (just describe what you have in your mind) - this cannot attain full marks, but it is still a perfectly acceptable answer and can get partial marks.*

*In essence, just submit what you have even if you don’t know the answer!*

A screenshot of a computer

Description automatically generated

def is\_palindrome(string):  
 start = 0  
 end = len(string) - 1  
  
 while start < end:  
 if string[start] != string[end]:  
 return False  
 start += 1  
 end -= 1  
  
 return True  
  
# Prompt the user to input a string  
user\_input = input("Enter a word to check if it's a palindrome: ")  
  
# This will call the is\_palindrome function with the user's input  
result = is\_palindrome(user\_input)  
  
# Display the result  
if result:  
 print("The word is a palindrome!")  
else:  
 print("The word is not a palindrome.")  
  
# Test cases

#user can comment out this code to ignore it and prevent it showing up in the terminal.  
stringA = "radar"  
print(is\_palindrome(stringA)) # Output: True  
  
stringB = "level"  
print(is\_palindrome(stringB)) # Output: True  
  
stringC = "Pencil"  
print(is\_palindrome(stringC)) # Output: False  
  
stringD = "a"  
print(is\_palindrome(stringD)) # Output: True

I chose to build this algorithm employing the following methods and steps:

1. I initialised two pointers, start and end, pointing to the first and last characters of the string, respectively.
2. I created a while loop where I set to compare the characters at start and end. If they don't match, my code would immediately return False as it would mean the word entered was not a palindrome.
3. If the characters at start and end match, the program will increment start and decrement end to continue comparing the next pair of characters.
4. The algorithm repeats steps 2-3 until start is no longer less than end, indicating that it has checked all the necessary pairs of characters.
5. If it hasn't returned False before reaching this point, it means all the characters matched, and the string is a palindrome, so it will return True.

The time complexity of this algorithm is O(n), n being the length of the input string. It needs to compare each character in the string once to determine if it is a palindrome. The space complexity is O(1) because it only uses a constant amount of additional space to store the two pointers. Therefore on both the time and the space complexity front it is a linear search algorithm which makes it an efficient code.O(1) is a constant form of algorithm that executes with the same amount of time regardless of the size of the data set (in this case the length of the word being checked), so it is efficient with any data set that you use it with. The time paradigm is slightly less efficient as you have o(n) linear – the linear algorithm whose performance declines in speed as the data set increases, however, I don’t think that would be an issue here as there are not many words that are more than multiple10s of digits long as opposed to numerical strings so this is still a good choice for the data format.

I think this algorithm is probably one of the most efficient solutions for determining palindromes. It iterates through the string, comparing characters from both ends, and terminates as soon as a mismatch is found. This way, the algorithm can handle palindromes efficiently without making unnecessary comparisons. So, it’s unlikely that a more efficient way to determine palindromes could be found, because the algorithm will always need to examine all characters in the string at least once.

|  |  |
| --- | --- |
| **Algorithms 2 (Coding)** | **20 MARKS**  **(1 question)** |

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] | “Nothing 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” |

*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 e cient way (and if so, how)?*

*If you are short on time, you can also submit pseudocode or simply describe what solution you’d write in code (just describe what you have in your mind) - this cannot attain full marks, but it is still a perfectly acceptable answer and can get partial marks.*

*In essence, just submit what you have even if you don’t know the answer!*

A screen shot of a computer

Description automatically generated

def find\_missing\_numbers(array):  
 found = set(array)  
 n = len(array) + 1  
 missing\_numbers = []  
  
 for num in range(1, n):  
 if num not in found:  
 missing\_numbers.append(num)  
  
 if len(missing\_numbers) == 0:  
 return "Nothing is missing"  
 elif len(missing\_numbers) == 1:  
 return f"Missing number: {missing\_numbers[0]}"  
 else:  
 return f"Missing numbers: {', '.join(str(num) for num in missing\_numbers)}"  
  
  
# Here I am Prompting the user to input an integer array  
user\_input = input("Enter an integer array (comma-separated): ")  
try:  
 arr = [int(num) for num in user\_input.split(',')]  
  
 # Checking for negative numbers as asked in the question  
 if any(num < 0 for num in arr):  
 print("Invalid input, negative number detected")  
 else:  
 result = find\_missing\_numbers(arr)  
 print(result)  
except ValueError:  
 print("Invalid input, non-integer value detected")

My solution has a time complexity of O(n) and a space complexity of O(n), with n being the length of the input array.

In terms of time complexity, the code iterates through the input array to check for negative numbers, which takes O(n) time. Then, the find\_missing\_numbers function iterates from 1 to n + 1 to identify the missing numbers, which also takes O(n) time. Therefore, the overall time complexity is O(n).

In terms of space complexity, the code employs a set called found to store the unique numbers from the input array, which requires O(n) space since the set may store all n numbers in the worst-case scenario. Additionally, the missing\_numbers list can store up to n numbers if all numbers are missing, resulting in O(n) space complexity. Thus, the overall space complexity is also O(n).

I chose this approach because it allows for efficient handling of the given requirements as there are many possible options that can be input by the user which as impossible to predict so there needs to be a large storage capacity and running time for the code to allow for this. The code checks for negative numbers before proceeding with further processing, so that an appropriate error message can be displayed when necessary. The find\_missing\_numbers function uses a set to efficiently determine missing numbers and constructs the output message accordingly.

This approach is already quite efficient, given the complexity of the possible user input options I can’t think of a significantly more efficient way to solve the problem in terms of time complexity. Since the solution requires examining each number at least once, I don’t think the time complexity cannot be improved beyond O(n), therefore additional examinations to, for example, identify negative numbers do not increase the complexity. My algorithm handles the task effectively by iterating through the input array and employing a set to track the presence of numbers.

However, theoretically, I could have approached it slightly differently if the requirement allows for a more space-efficient solution. Instead of using a set to track the presence of numbers, I could have used an array of booleans. This would reduce the space complexity to O(1) as we only need an array of size n to track the presence of n numbers. However, this approach would require assuming a maximum value for n (or the range of numbers), and it would only work if the given requirement allowed for such an assumption. So, more constraints on user input would be required to allow this approach to be viable which I didn’t have time for.