CSE331 HW2 REPORT

FUNCTIONS:

findBiggerIndexes:

- *Inputs:*
    1. source array
    2. meaningful length of source array (length which is not filled with empty index value)
    3. target element
    4. start index
- *Outputs:* numbers of bigger indexes with bigger values than target element
- *Explanation:* This function compares each word index of source array starting from start index to target element and checks if value of current index is bigger than target element and if it is, then stores the index in bigger_indexes array. This is done until iteration counter reaches the length.
- *Time Complexity:* Θ(n)
- *Space Complexity:* Θ(1)

findLength:

- *Inputs:* source array (array to find length)
- *Outputs:* meaningful length of source array (length which is not filled with empty index value)
- *Explanation:* This function checks each word index of source array if it is empty index value. If it is not, increments the counter by 1 and if it is, stops the counter and finishes the function.
- *Time Complexity:* O(1)
- *Space Complexity:* Θ(1)

clearArray:

- *Inputs:* source array (array to clear)
- *Outputs:* none
- *Explanation:* This function takes the source array and replaces each word of it with empty index value.
- *Time Complexity:* Θ(1)
- *Space Complexity:* Θ(1)

copyArray:

- *Inputs:*
    1. source array
    2. target array
- *Outputs:* none
- *Explanation:* This function first clears the target array, then saves each word of source array to the corresponding location in target array.
- *Time Complexity:* Θ(1)
- *Space Complexity:* Θ(1)

split:

- *Inputs:*
    1. source array
    2. target array
    3. length of source array (number of characters read)
- *Outputs:* number of integers split (including newline values)
- *Explanation:* This function goes through each byte of source array and splits integers between two comma characters and saves them in target array. If it encounters a newline character, places newline value to current index of target array to indicate that the part after this index is a new row.
- *Time Complexity:* Θ(n)
- *Space Complexity:* Θ(1)

findRowLength:

- *Inputs:*
    1. source array
    2. meaningful length of source array
    3. target array
- *Outputs:* none
- *Explanation:* This function goes through each word of source array and checks if it is newline value. If it is not, increments counter by 1 and if it is, saves the counter to current index of target array, resets the counter, and moves to the next index of target array.
- *Time Complexity:* Θ(n)
- *Space Complexity:* Θ(1)

printNewLine:

- *Inputs:* none
- *Outputs:* none
- *Explanation:* This function prints a new line character.
- *Time Complexity:* Θ(1)
- *Space Complexity:* Θ(1)

Main Function:

- *Inputs:* none
- *Outputs:* none
- *Explanation:* This is the main function that finds sequences.
- *Time Complexity:* $\Theta(n^3)$
- *Space Complexity:* Θ(1)

TEST CASE RESULTS:

Result of row-1:

```
candidate sequence: [26, 45, 81] size: 3
candidate sequence: [26, 81] size: 2
candidate sequence: [26, 44] size: 2
candidate sequence: [26, 36] size: 2
candidate sequence: [45, 81] size: 2
candidate sequence: [81] size: 1
candidate sequence: [44] size: 1
candidate sequence: [36] size: 1

result sequence: [26, 45, 81] size: 3
```

Result of row-2:

```
candidate sequence: [75, 89, 90, 93] size: 4
candidate sequence: [75, 90, 93] size: 3
candidate sequence: [75, 85, 93] size: 3
candidate sequence: [75, 93] size: 2
candidate sequence: [89, 90, 93] size: 3
candidate sequence: [89, 93] size: 2
candidate sequence: [90, 93] size: 2
candidate sequence: [85, 93] size: 2
candidate sequence: [26, 93] size: 2
candidate sequence: [26, 54] size: 2
candidate sequence: [1, 93] size: 2
candidate sequence: [1, 54] size: 2
candidate sequence: [93] size: 1
candidate sequence: [54] size: 1

result sequence: [75, 89, 90, 93] size: 4
```

Result of row-3:

```
candidate sequence: [34, 81, 88] size: 3
candidate sequence: [34, 88] size: 2
candidate sequence: [14, 81, 88] size: 3
candidate sequence: [14, 88] size: 2
candidate sequence: [5, 81, 88] size: 3
candidate sequence: [5, 88] size: 2
candidate sequence: [1, 81, 88] size: 3
candidate sequence: [1, 88] size: 2
candidate sequence: [81, 88] size: 2
candidate sequence: [88] size: 1

result sequence: [34, 81, 88] size: 3
```

Result of row-4:

```
candidate sequence: [18, 81, 96] size: 3
candidate sequence: [18, 65, 70, 96] size: 4
candidate sequence: [18, 26, 70, 96] size: 4
candidate sequence: [18, 70, 96] size: 3
candidate sequence: [18, 61, 96] size: 3
candidate sequence: [18, 96] size: 2
candidate sequence: [18, 42] size: 2
candidate sequence: [81, 96] size: 2
candidate sequence: [65, 70, 96] size: 3
candidate sequence: [65, 96] size: 2
candidate sequence: [26, 70, 96] size: 3
candidate sequence: [26, 61, 96] size: 3
candidate sequence: [26, 96] size: 2
candidate sequence: [26, 42] size: 2
candidate sequence: [70, 96] size: 2
candidate sequence: [61, 96] size: 2
candidate sequence: [96] size: 1
candidate sequence: [42] size: 1

result sequence: [18, 65, 70, 96] size: 4
```

Result of row-5:

```
candidate sequence: [26, 89] size: 2
candidate sequence: [26, 28] size: 2
candidate sequence: [89] size: 1
candidate sequence: [28] size: 1
candidate sequence: [14] size: 1

result sequence: [26, 89] size: 2
```

Result of row-6:

```
candidate sequence: [87, 93] size: 2
candidate sequence: [17, 76, 93] size: 3
candidate sequence: [17, 61, 93] size: 3
candidate sequence: [17, 93] size: 2
candidate sequence: [76, 93] size: 2
candidate sequence: [61, 93] size: 2
candidate sequence: [93] size: 1

result sequence: [17, 76, 93] size: 3
```

MISSING PART:

Writing the results to an output file is missing. Instead of that, program writes the results to the console.

Burak Yıldırım
1901042609