DAA - LAB - 4

TASK - 1:

Algorithm:

#CSV file format

serial number, [course-code-1, Course-code-2, Course-code-3, Course-code-4, course-code-5, course-code-6]

Assumption: Course code should only be numbers and we ~~cant~~ give error when it has anything else

```
//Input: Taken from csv file with above format
file_path = dataset.csv
dataset = readcsv (file_path)
```

```
Inversion(dataset)
// Input: a CSV file dataset of given format
// output: Dictionary with number of students
           for every inversion count needed.
count = {0:0, 1:0, 2:0, 3:0, 'others': 0}
for each student in dataset
    cc = All course codes of student as array
    invcount = CountInversions(cc)
    if invcount == 0
        count[0] ++
    else if invcount == 1
        count[1] ++
    else if invcount == 2
        count[2] ++
    else if invcount == 3
        count[3] ++
    else
        count['others'] ++
return count

CountInversions(arr)
// Input: An array of positive integers
// Output: Sorted array and Number of Inversions in array
if len(arr) <= 1:
    return arr, 0
mid = len(arr)/2
~~leftinv~~ left, leftinv = CountInversions(1st ~~2nd~~ half of array)
    right, rightinv = CountInversions(2nd half of array)
    merge, splitinv = MergeandCountSplit(left, right)
return merge, leftinv + rightinv + splitinv
```

```
MergeandCountSplit(left, right)
//Input: 2 arrays left & right
//Output: counts number of split inversions
//Outputs: Return sorted merged array & counts number of
           split inversions
result = []
i = j = split = 0
n = len(left)
while i < len(left) and j < len(right):
    if left[i] <= right[j]
        result.append(left[i])
        i++
    else
        result.append(right[j])
        j++
        split += len(left) - i
result.append(All elements right of left[i])
result.append(All element right of right[i])
return result, split
```

Test cases:

Ans.

- For positive testcases give appropriate answer
- for negative testcases:
  - Any of the course codes & has non numbers:
    Output: INVALID INPUT
  - If dataset is empty:
    Output: EMPTY FILE
  - If any column is left unfilled:
    Output: INVALID SIZE OF COLUMNS

- Positive testcases:
  - Input: student-course-codes-1
    Output: There are 0 students with 0 Inversions
    There are 4 students with 1 Inversion
    There are 3 students with 2 Inversions
    There are 10 students with 3 Inversion
    There are 83 students with Other Inversion
  - Input: student-course-codes-2
    Output: There are 2 students with 0 Inversions
    There are 3 students with 1 Inversion
    There are 10 students with 2 Inversions
    There are 15 students with 3 Inversions
    There are 70 students with Other Inversions

Task - 2:

Algorithm:

```
NormalMultiplication (x, y)
// Input: 2 Large Integers x & y
// Output: Product of x & y
n1 = len(x)
n2 = len(y)
if x == 0 or y == 0
    return 0
Initialise 'prod' string of size n1+n2 of 0s
for each digit i from right to left in x
    cy = 0
    for each digit j from right to left in y
      ~~result[i+j+1] = x[i] * y[j] + cy~~
      produ = x[i] * y[j]
      total = produt carry + prod[i+j+1]
      prod[i+j+1] = total % 10
      carry = total / 10
    prod[i+j] += cy
Remove leading zeros from ~~result~~ prod
return prod
```

Karatsuba (x, y)

// Input : 2 large integers x & y

// Output : product

if len(x) == 1 or len(y) == 1

    return x * y

m = Half the length of longer number

Split x into x_1 & x_0

Split y into y_1 & y_0

P_1 = Karatsuba (x_1, y_1)

P_2 = Karatsuba (x_0, y_0)

P_3 = Karatsuba ($x_1 + x_0$, $y_1 + y_0$)

Result = $P_1 * 10^{(2*m)} + (P_3 - P_1 - P_2) * 10^m + P_2$

return Result.

TESTCASES:

1. x = 10 , y = 10

Output: Product by Normal Multiplication is 100

Product by Karatsuba is 100

2. x = 111111111111 , y = 111111111111

Output: Product by Normal Multiplication is

12345678987654321

Product by Karatsuba is 12345678987654321

3. x = 15.967 , y = 100

Output: INVALID INPUT

4. x = abcd , y = 2

Output: INVALID INPUT

5. $x = 12345678987654321012$

$y = 21012345678987654321$

Output: Product by normal Multiplication is

25941167453040695067235848193782289
2852

Product by Karatsuba is.

259411674530406950672358481937822892852

TIME COMPLEXITY :

- TASK 1 :

— Brute Force :

- It has 2 nested loops without any termination
- ∴, Each loop runs for n iterations
- ∴, $T(n) \in O(n^2)$
- ∴, Time complexity is $O(n^2)$

— Divide & Conquer

- we divide array in 2 halves and then count split inversions in $O(n)$ time

s

- ∴, $T(n) = 2T(n/2) + O(n)$

By Master theorem

$a = 2, b = 2, d = 1$

$b^d = 2^1 = 2$

$a = b^d$

∴, $T(n) \in O(n \log n)$

- ∴, Time complexity is ~~n~~ $O(n \log n)$

Task 2:

— Brute Force:

- At Consider x & y to have same length of n
- ∴, we have 2 nested loops, looping from 0 to $n-1$,
- ∴, $T(n) = \sum_{i=0}^{n-1} \sum_{i=0}^{n-1} 1$

$$T(n) = n^2$$

∴, $T(n) \in O(n^2)$

- ∴, Time complexity is $O(n^2)$

— Divide & Conquer:

- ∴, There are 3 recursive calls for multiplication & addition takes $O(n)$ time
- ∴, $T(n) = 3T(n/2) + O(n)$
- By Master method,

$a = 3, \quad b = 2, \quad d = 1$

$b^d = 2^1 = 2$

$a > b^d$

∴, $T(n) \in O(n^{\log_2 3})$

$T(n) \in O(n^{1.585})$

- ∴, Time complexity is $O(n^{1.585})$