

**Indian Institute of Technology Roorkee
Department of Computer Science and Engineering**

CSN-261: Data Structures Laboratory (Autumn 2019-2020)

Lab Assignment – 2

Name: Hemil Sanjaybhai Panchiwala

Enrollment No.: 18114031

Branch: Computer Science & Engineering

Sub Batch: O2



Problem Statement 1

In this Problem, you have to implement a simple transposition cipher, where this cipher encrypts and decrypts a sequence of characters by dividing the sequence into blocks of size n , where n is specified by the encryption key. If the input text has a length that is not a multiple of n , the last block is padded with null characters ('\0').

In addition to n , the key also specifies two parameters a and b . For each block, the i -th output character, starting from 0 as usual, is set to the j -th input character, where $j = (a_i + b) \bmod n$. For appropriate choices of a and b , this will reorder the characters in the block in a way that can be reversed by choosing a corresponding decryption key (n, a', b') .

Data Structures used :-

Array

Algorithm

$$j = (ai + b) \% n$$

$$i = (a'j + b') \% n$$

$$ai + b = q_1 n + j$$

$$a'j + b' = q_2 n + i$$

$$a'(ai + b - q_1 n) + b' = q_2 n + i$$

$$aa'i + a'b - aq_1 n + b' = q_2 n + i$$

$$i(aa' - 1) + (a'b + b') = n(q_2 + aq_1)$$

So, as $q_2 + aq_1$ is integer,

$$\text{So, } (aa' - 1) \% n = 0$$

$$aa' \% n = 1$$

So a' is modular inverse of a .

$$(a'b + b') \% n = 0$$

$$\therefore b' = n - ((a'b) \% n)$$

Snapshots

- Test Case 1:

```
blackreaper@blackreaper:~/Desktop
$ g++ transpose.c
blackreaper@blackreaper:~/Desktop
$ ./a.out 5 3 2 inputfile.txt
mSpaltee scta sces
n6-12*
blackreaper@blackreaper:~/Desktop
$ g++ inverseTranspose.c
blackreaper@blackreaper:~/Desktop
$ ./a.out 5 3 2 outputfile.txt
Sample test case
csn-261
```

```
inputfile.txt x
inputfile.txt
1 Sample test case
2 csn-261
```

```
outputfile.txt x
outputfile.txt
1 mSpaltee scta sces
2 n6-12*
```

```
decryptedOutputfile.txt x
decryptedOutputfile.txt
1 Sample test case
2 csn-261
```

```
blackreaper@blackreaper:~/Desktop/Desktop Folder/dslabs Assignments/Assignment 2$ g++ compareFiles.c
blackreaper@blackreaper:~/Desktop/Desktop Folder/dslabs Assignments/Assignment 2$ ./a.out
Your program shows correct output
```

Time taken: transpose.c -> 0.000562 s

inverseTranspose.c -> 0.000706 s

compareFiles.c -> 0.000262 s

- Test Case 2:

```
blackreaper@blackreaper:~/Desktop/D
$ g++ transpose.c
blackreaper@blackreaper:~/Desktop/D
$ ./a.out 5 3 2 inputfile.txt
mSpaltee scta s
ec2s2n6-1
blackreaper@blackreaper:~/Desktop/D
$ g++ inverseTranspose.c
blackreaper@blackreaper:~/Desktop/D
$ ./a.out 5 3 2 outputfile.txt
Sample test case2
csn-261
```

```
≡ inputfile.txt ×
≡ inputfile.txt
1 | Sample test case2
2 | csn-261
```

```
≡ outputfile.txt ×
≡ outputfile.txt
1 | mSpaltee scta s
2 | ec2s2n6-1
```

```
≡ decryptedOutputfile.txt ×
≡ decryptedOutputfile.txt
1 | Sample test case2
2 | csn-261
```

```
blackreaper@blackreaper:~/Desktop/Desktop Folder/dslabs Assignments/Assignment 2$ g++ compareFiles.c
blackreaper@blackreaper:~/Desktop/Desktop Folder/dslabs Assignments/Assignment 2$ ./a.out
Your program shows correct output
```

Time taken: transpose.c -> 0.000167 s

inverseTranspose.c -> 0.000466 s

compareFiles.c -> 0.000274 s

Problem Statement 2

Medial axis transformation (MAT)

Write a C program, MAT.c to represent any region (in image array representation), into its quadtree form.

Input: Sample region is represented as $n \times n$ array (as shown in Fig. 1 using 6×6 matrix).

The format of the input file should be as follows:

the pixel values in the input file are separated by a single space and rows are separated by a newline character (refer to the sample L2_P2_inputsample.txt file shared in Piazza).

Output:

1. Print the Maximal square array where it should be filled in the following order: top-right, top-left, bottom-right and bottom-left quadrant, this should be done recursively for all the sub-quadrants. All the cells within a maximal square block should be filled with its corresponding block number.
2. Print the quadtree in the following manner, labels of leaf nodes, corresponding bit value and their level information (assuming the level of the root node to be 0), while traversing the quadtree in postorder. For example, in Fig. 2(d) the leaf node 3 having bit value 0 at level 2 and should be printed as (3,0,2).

Data Structures used :-

Array

Algorithm

- **In this code, I have taken the input array from the txt file and then I have checked whether all elements of array are same or not.**
- **If same, then make all indices of same value else break array into four parts and check again and do this process till all the indices are checked completely i.e. when the array becomes maximal square array.**
- **Then I have made a quadtree with each node having three values as its level, label and indexValue.**
- **Then I have printed the maximal square array and the quadtree with all its three values.**

Snapshots

```
blackreaper@blackreaper:~/Desktop/Desktop Folder/dslabs Assignments/Assignment 2$ g++ MAT.c
blackreaper@blackreaper:~/Desktop/Desktop Folder/dslabs Assignments/Assignment 2$ ./a.out
Maximal Square Array:
1 1 1 1 2 2 3 3
1 1 1 1 2 2 3 3
1 1 1 1 4 4 5 5
1 1 1 1 4 4 5 5
6 6 7 8 13 13 14 14
6 6 9 10 13 13 14 14
11 11 12 12 15 16 19 19
11 11 12 12 17 18 19 19

Quad Tree:
(indexValue, value, level)
1 0 1
2 0 2
3 0 2
4 1 2
5 1 2
6 0 2
7 0 3
8 1 3
9 1 3
10 1 3
11 0 2
12 1 2
13 1 2
14 1 2
15 1 3
16 1 3
17 1 3
18 0 3
```

Time taken: 0.000680 s