ASSIGNMENT 4

Name: Imon Raj
Class: BCSE III
Roll: 002010501098
Section: A3
Subject: Computer Networks Lab Report

PROBLEM STATEMENT: In this assignment you have to implement CDMA for multiple access of a common channel by n stations. Each sender uses a unique code word, given by the Walsh set, to encode its data, send it across the channel, and then perfectly reconstruct the data at n stations

DESIGN: This problem was quite easy to understand and implement. CDMA stands for Code Division Multiple Access and it is one of the MAC protocols. Here all n stations in a shared media can use the complete bandwidth anytime. Here the trick is that, each station will have a code associated with it. Any two such codes will be orthogonal to each other. These codes for stations can be generated using WalshTable.

The station which sends a bit, will do a calculation. It will consider 0 bit as -1 and 1 as 1. Then, if it's code is C_i, then C_i is multiplied with the bit value. The generated result is sent as signal sequence onto the media. If any $station_i$ wants to receive data from $station_j$ then, $station_i$ will get signal values from the media and multiply then with the code C_j. Then the result is divided by no of stations to get the bit sent by $station_j$.

Generation of Walsh Table is interesting. We will start with a (2 x 2) walshTable and populate it to our desired size. One thing to note that, for any given station count n, we have to calculate a K>=n such that K=2^p. Now we will populate the table to size (K x K). Surely, we will have K codes, we will only use n of them for n stations.

IMPLEMENTATION:

All related codes written in Java are attached herewith (In the Drive-shared-folder).

TEST CASES:

- The K for a given n must be calculated properly.
- The system should be able to detect correctly bit 0, 1 and no-bit receiving.
- walshTable generation should be as time efficient as possible.

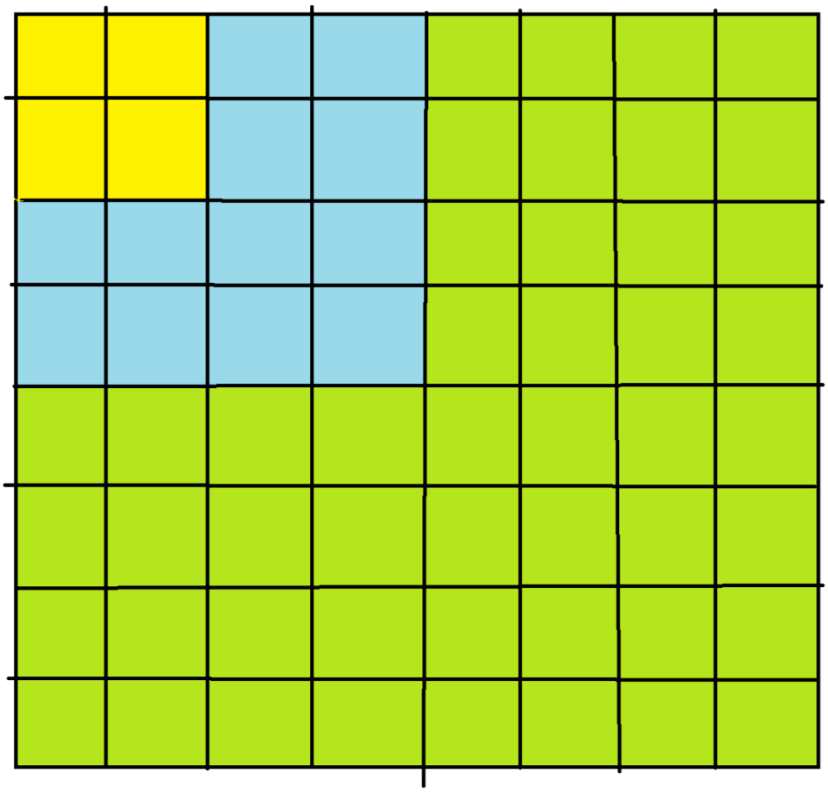
One sample output is given below –

```
Enter the number of stations:
13

----------WALSH TABLE---------
1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1
1       -1      1       -1      1       -1      1       -1      1       -1      1       -1      1       -1      1       -1
1       1       -1      -1      1       1       -1      -1      1       1       -1      -1      1       1       -1      -1
1       -1      -1      1       1       -1      -1      1       1       -1      -1      1       1       -1      -1      1
1       1       1       1       -1      -1      -1      -1      1       1       1       1       -1      -1      -1      -1
1       -1      1       -1      -1      1       -1      1       1       -1      1       -1      -1      1       -1      1
1       1       -1      -1      -1      -1      1       1       1       1       -1      -1      -1      -1      1       1
1       -1      -1      1       -1      1       1       -1      1       -1      -1      1       -1      1       1       -1
1       1       1       1       1       1       1       1       -1      -1      -1      -1      -1      -1      -1      -1
1       -1      1       -1      1       -1      1       -1      -1      1       -1      1       -1      1       -1      1
1       1       -1      -1      1       1       -1      -1      -1      -1      1       1       -1      -1      1       1
1       -1      -1      1       1       -1      -1      1       -1      1       1       -1      -1      1       1       -1
1       1       1       1       -1      -1      -1      -1      -1      -1      -1      -1      1       1       1       1
---------------------------
Station 0 generates no data.
Station 1 generates no data.
Station 2 generates data bit 1
Station 3 generates data bit 0
Station 4 generates data bit 1
Station 5 generates no data.
Station 6 generates no data.
Station 7 generates no data.
Station 8 generates data bit 1
Station 9 generates data bit 0
Station 10 generates data bit 0
Station 11 generates data bit 0
Station 12 generates data bit 1
---------------------------
Channel Data:
0       6       4       2       -4      2       0       -2      2       0       -2      -4      2       0       -2      -4
---------------------------

Which station you want to see:
4
Data bit from station 4 is : 1
```

RESULTS & ANALYSIS:

Program can fulfil all the test cases conditions provided above. Currently, I haven't found any bug, as the program is not very complicated.

Let's analyse the Time complexity of the walshTable generation function. Notice carefully the image given below –



At the beginning we will have only the yellow portion's code. Using this yellow portion we will calculate all the blue portions, then using blue and yellow we will generate the green portion and so on. Now for any given N(2^p form) time complexity will be –

$T(N) = 3*(4 + 16 + 64 + \ldots\ldots\ldots + N^2)$

$= 3*(2^2 + 4^2 + 8^2 + \ldots\ldots\ldots..+ N^2)$

$= 3*4*(1^2 + 2^2 + 4^2 + \ldots\ldots\ldots..+ (N/2)^2)$

$= 12 * 1 * (4^p - 1)/(4 - 1)$ [put N = 2^p]

$= 4 * (4^p - 1)$

$= 4 * (N^2 - 1)$

$= \mathbf{O(N^2)}$

So, we can see the time complexity is O(N^2). We can even think easily that each slot in that matrix is calculated once, so we have total N*N operations, which leads us to the same result.

COMMENTS:

This assignment was not that lengthy. Like all the other assignments, it also encouraged me to think of an efficient solution. And coding this assignment was mostly fun and interesting.