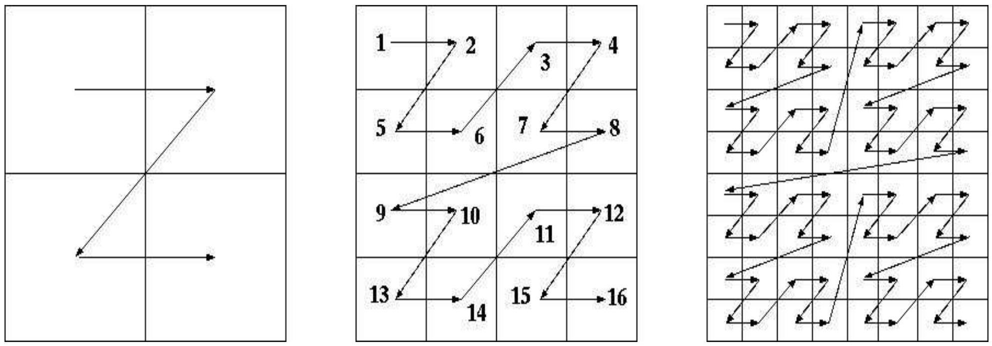
1. Consider the following Z curve traversal of an 2D array of integers. Given a set of $n \times n = n^2$ integers stored in square matrix of size $n \times n$. Your task is to traverse the elements in the matrix along the Z curve. In the given example, if the matrix contains the elements $1, 2, 3, 4, 5, 6, 7, 8, \ldots, 13, 14, 15, 16$, then the Z curve traversal will give the output: $1, 2, 5, 6, 3, 4, 7, 8, 9, 10, 13, 14, 11, 12, 15, 16$.



(a) Write a divide-and-conquer recursive algorithm for the above task for $n \neq 2^k$, for some $k \geq 0$. Consider the algorithm zee, where the matrix contains elements; row_begin, row_end are indices for begin and end of the row in the matrix, and col_begin, col_end are indices for begin and end of the column in the matrix:

zee (row_begin, row_end, col_begin, col_end, matrix)

{

⋮

}

Compute the time complexity for the algorithm, zee().

(b) Extend your approach for $n \neq 2^k$, for some $k \geq 0$.

[(5+2) + 3 = 10]

Solution:

(a) The following is the algorithm for the above task.

Algorithm 1 zee(row_begin, row_end, col_begin, col_end, matrix)

```
{The matrix contains elements; row_begin, row_end are indices for begin and end of the row in the matrix,
and col_begin, col_end are indices for begin and end of the column in the matrix}
{Small problem when the submatrix will have only a single element}
if ((row_begin = row_end) and (col_begin = col_end)) then
   print element matrix[row_begin][col_begin];
   return ;
else
   {Divide the problem into four smaller subproblems}
   zee(row_begin, (row_begin+row_end)/2, col_begin, (col_begin+col_end)/2, matrix );
   zee(row_begin, (row_begin+row_end)/2, (col_begin+col_end)/2 + 1, col_end, matrix );
   zee((row_begin+row_end)/2 + 1, row_end, col_begin, (col_begin+col_end)/2, matrix );
   zee((row_begin+row_end)/2 + 1, row_end, (col_begin+col_end)/2 + 1, col_end, matrix );
end if
```

□

Let $T(n)$ be the time complexity required for the Z curve traversal, where n be the order of the matrix. Then, from the above algorithm it follows that

$$\begin{aligned} T(1) &= 1 \\ T(n) &= 4T(n/2) + 1, \text{ if } n > 1. \end{aligned}$$

Thus, $T(n) = O(n^{\log_2 4}) = O(n^2)$.

□