1 Homework 7

Preparatory to developing a "full-blown" multi-level FMM code, we will develop several functions that will be necessary for doing book-keeping for the FMM. Create a library of functions that enable you to work with a hierarchical data in 2^d- tree. Minimum requirements include case $d = 1$ and the following functions with the specified complexity, output arguments, name, and input arguments.

Complexity	Output	$= Function(arg)$	Description
$O(1)$	m	$= Parent(n)$	n is the box index, m is the parent index
$O(1)$	Ch	$= ChildrenAll(n)$	n is the box index
			Ch is the array of children box indices
$O(1)$	Sb	$= SiblingsAll(n)$	n is the box index
			Sb is the array of sibling box indices
$O(1)$	Nei	$= NeighborsAll(n, l)$	n is the box index at level l
			Nei is the array of neighbor box indices
$O(1)$	$NeiE4$	$= NeighborsE4All(n, l)$	n is the box index at level l, $NeiE4$ is the array of box indices
			at the same level that belong to the $E4$-neighborhood
$O(1)$	n	$= BoxIndex(x, l)$	x is the point coordinate in a unit cube
			n is the index of box at level l containing x
$O(1)$	x	$= BoxCenter(n, l)$	n is the box index at level l
			x is the point coordinate in a unit cube
$O(1)$	s	$= BoxSize(l)$	l is level, s is the size of the box at this level
$O(\log N)$	Ch	$= Children(n, l, X)$	n is the box index at level l
			Ch is the array of nonempty children box indices
$O(\log N)$	Sb	$= Siblings(n, l, X)$	n is the box index
			Sb is the array of nonempty sibling box indices
$O(\log N)$	Nei	$= Neighbors(n, l, X)$	n is the box index at level l
			Nei is the array of nonempty neighbor box indices
$O(\log N)$	$NeiE4$	$= NeighborsE4(n, l, X)$	n is the box index at level l, $NeiE4$ is the array of nonempty box
			indices at the same level that belong to the $E4$-neighborhood
$O(N)$	L	$= GetMaxLevel(s, X)$	X is sorted data array, L is the space subdivision level
			at which each box contains not more than s points

You also may write a function $SetDataStructure(s, X)$ (that requires $O(N \log N)$ time) which will be useful for further work. It takes an initial array X and grouping parameter s, and orders the array, obtains its permutation index, determines maximum level of subdivision required, and creates nesessary data for FMM bookkeeping.

1. Implement these functions using bitwise operations in Matlab;
2. Make tests and check if your programs work correctly (take some small level and data set of small length to check this manually).
3. Print out and submit the following results for 1-d case: $Parent(1025)$, $ChildrenAll(1025)$, $SiblingsAll(1025)$, $NeighborsAll(1025, 12)$, $NeighborsE4All(1025, 12)$, $BoxIndex(0.1234567, 15)$, $BoxCenter(1025, 12)$, $BoxSize(15)$.
4. Generate a data set X that contains 2000 numbers equispaced in [0,0.9995]. Print out and submit the following results for 1-d case $Children(125, 9, X)$, $Siblings(1025, 11, X)$, $Neighbors(1025, 11, X)$, $NeighborsE4(1025, 11, X)$.
5. Print out and submit for this X result of call $GetMaxLevel(7, X)$.

Hints

1. For debugging and tests try first to draw on a paper a picture that tells you what numbers you should get.