Homework 7 (Data Structures)

Create a library of functions that enable you to work with hierarchical data on a 2^d- tree. Your program should at least be able to handle the cases $d = 1,2$. (It is relatively straightforward to create a library, which takes d as an input parameter).

The library must contain the following functions

Bit interleaving/deinterleaving:

$n = Interleave(N)$, $\quad$ {n is the box index in the 2^d-tree, N is the vector of length d
(integer Cartesian coordinates of n, $N_k = 0, 1, ..., k = 1, ..., d$)},
$N = Deinterleave(n)$, {the same as above}.

Functions that require constant time $O(1)$:

$$
\begin{aligned}
m &= Parent(n), \quad \{n \text{ is the box index, } m \text{ is the parent index}\}, \\
Ch &= ChildrenAll(n), \quad \{n \text{ is the box index, } Ch \text{ is the array of children box indexes}\}, \\
Sb &= SiblingsAll(n), \quad \{n \text{ is the box index, } Sb \text{ is the array of sibling box indexes}\}, \\
Nei &= NeighborsAll(n,l), \quad \{n \text{ is the box index at level } l, \\
& \qquad Nei \text{ is the array of neighbor box indexes}\}, \\
NeiE4 &= NeighborsE4All(n,l), \quad \{n \text{ is the box index at level } l, NeiE4 \text{ is} \\
& \qquad \text{the array of box indexes at the same level that belong to the E4-neighborhood}\}, \\
n &= BoxIndex(x,l), \quad \{x \text{ is the point coordinate in a unit cube,} \\
& \qquad n \text{ is the index of box at level } l \text{ containing } \mathbf{x}\}, \\
x &= BoxCenter(n,l), \quad \{n \text{ is the box index at level } l, x \text{ is the point coordinate in a unit cube}\}, \\
s &= BoxSize(l), \quad \{l \text{ is level, } s \text{ is the size of the box at this level}\}.
\end{aligned}
$$

Functions that require time $O(\log N)$

These functions will need to use the function $find$. The term non-empty means that the box contains at least one point of the sorted data set X

$$
\begin{aligned}
Ch &= Children(n,l,X), \quad \{n \text{ is the box index, } Ch \text{ is the array of nonempty children box indexes}\}, \\
Sb &= Siblings(n,l,X), \quad \{n \text{ is the box index, } Sb \text{ is the array of nonempty sibling box indexes}\}, \\
Nei &= Neighbors(n,l,X), \quad \{n \text{ is the box index at level } l, \\
& \qquad Nei \text{ is the array of nonempty neighbor box indexes}\}, \\
NeiE4 &= NeighborsE4(n,l,X), \quad \{n \text{ is the box index at level } l, \\
& \qquad NeiE4 \text{ is the array of nonempty box indexes at the same level that belong to the E4-neighborhood}\},
\end{aligned}
$$

Function that requires $O(N)$ time:

$$L \;=\; GetMaxLevel(s, X), \qquad \{X \text{ is sorted data array, } L \text{ is the space subdivision level}$$
at which each box contains not more than s points\};

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

Questions

1. Learn the following Matlab functions (use help): `bitmax, bitand, bitor, bitxor, bitcmp, bitshift, bitset, bitget; sort, find, union, intersect, difference, setxor;`

2. Implement these functions using bitwise operations in Matlab;

3. Make tests and check if your programs work correctly (take some small level and data set of small length to check this manually).

4. Print out and submit the following results for 1-d case: `Parent(1025)`, `ChildrenAll(1025)`, `SiblingsAll(1025)`, `NeighborsAll(1025, l)`, `NeighborsE4All(1025, l)`,
   $BoxIndex(0.1234567, l), BoxCenter(1025, l), BoxSize(15)$, where $l = 12$ for $d = 1$ and $l = 6$ for $d = 2$.

5. 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(1025, 12, X)`, `Siblings(1025, 12, X)`, `Neighbors(1025, 12, X)`,
   `NeighborsE4(1025, 12, X)`.

6. Print out and submit for this X result of call $GetMaxLevel(7, X)$.

7. Generate a data set X that contains 10000 numbers equispaced in [0,0.99]×[0,0.99]. Print out and submit the following results for 2-d case `Children(1025, 6, X)`, `Siblings(1025, 6, X)`, `Neighbors(1025, 6, X)`, `NeighborsE4(1025, 6, X)`.

8. 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 which tells you what numbers you should get.