Data Structure I: Tutorial 4

Definition 1 (Search tree) *A search tree is a tree in which every node u satisfies:*

- *Node u has arbitrary many children.*
- *Node u with k children (pointers) has $k-1$ elements sorted by their keys.*
- *The i-th key of u is greater than all keys in the i-th subtree and smaller than all keys in the $(i+1)$-th subtree for every key i.*
- *A child without subtree is NULL pointer.*

Definition 2 ((a,b)-tree) *(a,b)-tree is a search tree satisfying the following conditions:*

- *a, b are integers such that $a \geq 2$ and $b \geq 2a-1$.*
- *All nodes except the root have at least a and at most b children.*
- *The root has at most b children.*
- *All nodes with a child equals NULL are at the same depth.*

In literature, (a,b)-trees have many variants called B-trees, B*-trees, B+-trees, 2-3-trees, etc.

Exercise 3 *Describe operations Find, Insert, and Delete in (a,b)-trees.*

Exercise 4 *If $b \geq 2a$, describe operations Insert and Delete which traverses the tree only from the root to a leaf (without traversing it back to the root).*

Exercise 5 *The number of children of a node is between a and $b \geq 2a-1$. So, if a node has only a children, approximately half of its memory is unused. Is it possible to modify operations Insert and Delete so that at most $1/3$ of nodes' memory is unused? I.e. describe operations in a variant of (a,b)-tree where $a \approx \frac{2}{3}b$.*