9.1. Given a tree, a number is written on each edge. Answer queries «find the sum on the path from v to u» in $O(\log n)$ time (well, or even in $O(1)$).

9.2. Given a tree, a number is written on each edge. Answer queries «find the minimum on the path from v to u» in $O(\log n)$ time.

9.3. Learn to calculate any associative function on a path in a tree in $O(\log n)$ time.

9.4. Given a tree, each edge has a non-negative length. Learn to answer queries: «find the closest ancestor of v, the distance to which is at least l» in $O(\log n)$.

9.5. Given a tree, each edge has a length. Learn to answer queries: «find the middle of a path from v to u» in $O(\log n)$ time (the middle of a path is either a vertex or a point on an edge, in the second case you can just find edge).

9.6. Queries: «given a set of k vertices, find their lowest common ancestor» in $O(k \log n)$.

9.7. Queries: «given a set of k vertices, find the number of vertices that are the ancestor of at least one of them» $O(k \log n)$.

9.8. Let the tree change sometimes. Let's add two operations: «create a new vertex v and connect it to u» and «remove a leaf v». Show that you can recalculate binary liftings without sacrificing structure performance.

9.9. Queries: 1) calculate the LCA of two vertices and 2) detach the subtree of the vertex u from its curent parent and connect it as a child to vertex v as child (both in $O(\log n)$).