10.1. What is the maximum and minimum possible number of paths in the Heavy-light tree decomposition of tree of size n?

10.2. There is another way to build Heavy-light decomposition: mark edges (u, v) as heavy, for which $size(v) \geq size(u)/2$. Show that this method also works and has the same asymptotics. Will it be better or worse than the method that selects the largest child?

10.3. We will choose an edge leading to a subtree of maximum height (not maximum weight) as a heavy edge. What is the maximum number of light edges that can be on the way to the root?

10.4. Learn how to process queries in polylog time: 1) change the length of an edge, 2) find the leaf farthest from the root.

10.5. There is a tree, each edge can be turned on or off. Learn how to process such queries in polylog time: 1) change the state of all edges on the path from u to v, 2) find the number of connected components by the edges turned on.

10.6. There is a tree, each edge can be turned on or off. Learn how to process such queries in polylog time: 1) change the state of an edge, 2) find the longest path from the root along the edges turned on.

10.7. There is a tree, vertices can be turned on and off. Requests: 1) turn off the vertex (it will never be turned on), 2) find the nearest turned on ancestor for the given vertex. Faster than $O(\log n)$.

10.8. There is a tree of roads, peak 1 is the capital, there are good and bad roads. Find the minimum number of roads to be repaired so that there is no more than one bad road $O(n)$ on the way from the capital to any vertex.

10.9. There is a two-track tree-like railway network. For each track, the time it takes for the train to pass is known. You need to answer the queries: «The first train leaves from A to B at time X, the second — from C to D at time Y, is it true that there is a point in time at which trains travel along the same edge?», in $O(\log n)$ time.