7.1. Show that if you make splay operation using only zig every time, then there is a sequence of n operations that take longer than $n \log n$ time.

7.2. Let the splay tree contain numbers from 1 to n. We will do m operations: `splay(1)`, `splay(n)`, `splay(1)`, `splay(n)`, ... What will be the total time of all operations?

7.3. In a splay tree of size n, you perform many operations on a small subset of k elements. What will the tree look like? How long will the operations work?

7.4. Let the AVL tree contain numbers from 1 to n. Show that if you do the find operation in order for all numbers from 1 to n, then the total running time will be more than $O(n)$.

7.5. Let the splay tree contain numbers from 1 to n. M queries are made to the tree, the total number of queries to i-th element is p_i. Show that the total time complexity of all queries is $O(m + \sum p_i \log \frac{m}{p_i})$ (hint: you need to choose the correct $w(v)$ so that $r(x)$ is as need to).

7.6. Show how to do the `split` operation in the splay tree. How long will will it work? (do not forget to calculate the change in potential).

7.7. Show how to do the `merge` operation in the splay tree. How long will will it work? (do not forget to calculate the change in potential).

7.8. Show how to build a splay tree from a given sorted array in $O(n)$ amortized time. (do not forget to calculate the change in potential).