RMQ, LCA

Problems

1. Use Sparse Table idea to solve static RMQ problem...
   (a) ...in $\langle \mathcal{O}(n \log\log n), \mathcal{O}(1)\rangle$ time.
   (b) ...in $\langle \mathcal{O}(n), \mathcal{O}(\log\log n)\rangle$ time.
   (c) ...in $\langle \mathcal{O}(n \log^* n), \mathcal{O}(\log^* n)\rangle$ time. What is $\log^* n$?

2. You are given a full binary tree of size $2^h - 1$ labelled in level order fashion. I.e. the root has label 1, and for each non-leaf vertex labelled i its left and right children has labels $2i$ and $2i+1$. Find a simple way to answer $\mathrm{LCA}(u, v)$ queries in this tree in $\mathcal{O}(1)$ time.
   (a) How the binary representation of label i and the path from root to i are related?
   (b) Solve the problem if $\mathtt{depth}(u) = \mathtt{depth}(v)$.
   (c) Solve the problem in general case.

3. Bender-Farach-Colton algorithm takes $\mathcal{O}(\sqrt{n}\log^2 n)$ time and space to precompute answers for all short ± 1 sequences. Show how to do that in just $\mathcal{O}(\sqrt{n})$ time and space, keeping the same $\frac{1}{2}\log_2 n$ block size and $\mathcal{O}(1)$ query time.

4. Create a data structure which starts with a single root vertex and can quickly answer the following queries: (1) add a new leaf vertex and (2) find $\mathrm{LCA}(u, v)$, both in $\mathcal{O}(\log n)$ time. You can use $\mathcal{O}(n \log n)$ memory.

5. You are given a tree where each vertex has an assigned value. Create a data structure which finds minumum value on a shortest path between a pair of vertices u, v in $\langle \mathcal{O}(n \log n), \mathcal{O}(\log n)\rangle$ time.

6. You a given a tree where each vertex can be colored or not.Create a data structure which can quickly answer the following queries: (1) color/uncolor any vertex and (2) find the number of colored vertices in a subtree, both in $\mathcal{O}(\log n)$ time. You can use $\mathcal{O}(n)$ memory.

7. You are given a binary operation $\circ : X \times X \to X$ which is associative ($(a \circ b) \circ c = a \circ (b \circ c)$), but not necessarily idemponent. You are asked to answer range queries $a_L \circ a_{L+1} \circ \ldots \circ a_{R-1}$ in $\mathcal{O}(1)$ time. Construct a data structure which allows that after $\mathcal{O}(n \log n)$ preprocessing.
   (a) Solve the special case $R - L \geqslant N/2$ with $\mathcal{O}(n)$ preprocessing.
   (b) Solve the general case for $N = 2^k$.

Compiled by Dmytro Korduban. Last updated: March 19, 2019