Mandatory Exercise: Predecessor, RMQ and LCA

Philip Bille Inge Li Gørtz

1 Labeled Predecessor and First Covering Ancestor Consider the following two problems.

Labeled Predecessor Let $S = \{0, \ldots, n-1\}$ be a set of integers. We say that S is *labeled* if each integer is associated with a *label* from a set of labels $L = \{0, \ldots, l-1\}$. Given a labeled set S, an integer x, and a label ℓ, a *labeled predecessor query* is defined as follows.

- label-predecessor(x, ℓ): return the largest element in S with label ℓ that is smaller than x.

Given a labeled set S, the *labeled predecessor problem* is to preprocess S into a compact data structure that supports labeled predecessor queries.

First Covering Ancestor Let T be a rooted tree with n nodes. We say that T is *labeled* if each leaf is associated with a *label* from a set of labels $L = \{0, \ldots, l-1\}$. Given a node $v \in T$, the subtree rooted at v, denoted $T(v)$, is the tree consisting of v and all descendants of v. A node $v \in T$ *covers a label* ℓ if $T(v)$ contains a leaf labeled ℓ. Given a leaf $v \in T$ and a label $\ell \in L$, a *first covering ancestor query* is defined as follows.

- FCA(v, ℓ): return the deepest ancestor a of v such that a covers ℓ.

Given a labeled tree T, the *first covering ancestor problem* is to preprocess T into a compact data structure that supports first covering ancestor queries.

Solve the following exercises.

1.1 Give a data structure for the labeled predecessor problem that answers queries in $O(1)$ time and uses little space. *Hint:* a good solution depends on both on the number of labels and n.

1.2 Give a data structure for the labeled predecessor problem that uses $O(n)$ space and supports fast queries.

1.3 Give a linear-space data structure for the first covering ancestor problem that supports fast FCA queries.

Ignore preprocessing in all of the exercises.