Data Structures and Algorithms

Red-Black Trees

Glossary

Here are some terms that are used when discussing *red-black* trees:

- **child** — same as for binary search trees
- **parent** — same as for binary search trees
- **leaf** — same as for binary binary search trees
- **grandparent** — parent of parent
- **uncle** — sibling of parent
- **niece** — closest child of sibling - if you are a right child, your niece is a right child - if you are a left child, your niece is a left child
- **nephew** — furthest child of sibling - if you are a right child, your nephew is a left child - if you are a left child, your nephew is a right child
- **black height** — the number of black nodes encountered on the way to a leaf - sometimes abbreviated *BH*
- **linear** — true if the parent and child are both left children or are both right children
- **rotation** — make a child a parent and the former parent a child - preserves binary search tree ordering

Note that terms like *uncle*, *niece*, and *nephew* are not found in other binary search tree descriptions. They are used here in order to remove leftness and rightness issues from the main red-black tree algorithms.

Red-black properties

A *red-black* tree has the following properties:

- The root is colored black
- The null child pointers of a leaf are considered black nodes
- No red node has a red parent
- All nodes have a consistent black height (all paths from a node to the reachable leaves have the same number of black nodes)

Next: *Inserting into red-black trees*