Bottom-Up Red-Black Trees

- Top-down red-black trees require $O(\log n)$ rotations per insert/delete.
- Color flips cheaper than rotations.
- Priority search trees.
  - Two keys per element.
  - Search tree on one key, priority queue on other.
  - Color flip doesn’t disturb priority queue property.
  - Rotation disturbs priority queue property.
  - $O(\log n)$ fix time per rotation => $O(\log^2 n)$ overall time.

Bottom-Up Red-Black Trees

- Bottom-up red-black tree properties.
  - At most 1 rotation per insert/delete.
  - O(1) amortized complexity to restructure following an insert/delete.

Bottom-Up Insert

- New pair is placed in a new node, which is inserted into the red-black tree.
- New node color options.
  - Black node => one root-to-external-node path has an extra black node (black pointer).
    - Hard to remedy.
  - Red node => one root-to-external-node path may have two consecutive red nodes (pointers).
    - May be remedied by color flips and/or a rotation.