Persistent Data Structure

Boyu Tian

ACM class
Shanghai Jiao Tong University

anshantby@163.com

July 15, 2014
Overview

Persistent Data Structure

Persistent Singly Linked Lists

Persistent Binary Trees

Persistent Balanced Trees
Persistent vs. Immutable

- An **immutable** data structure is one that, once created, cannot be modified
  - Immutable data structures can (usually) be copied, with modifications, to create a new version
  - The modified version takes up as much memory as the original version

- A **persistent** data structure is one that, when modified, retains both the old and the new version
  - Persistent data structures are effectively immutable, in that prior references to it do not see any change
  - Modifying a persistent data structure may copy *part* of the original, but the new version shares memory with the original
Why persistent data structures?

- Functional programming is based on the idea of immutable data, or persistent data, which is effectively immutable.
- Synchronization is expensive, and immutable data structures don’t need to be synchronized.
- Copying large data structures is expensive and wastes space, but persistent data structures can use sophisticated structure sharing to reduce the cost on disk between program executions.
Singly Linked Lists
Persistent Singly Linked Lists
Persistent Binary Trees
Persistent Binary Trees

Diagram of a persistent binary tree with nodes labeled a, b, c, d, e, f, g, g', h, d', xs, and ys.
Persistent Treap

- merge($a, b$)
- split($a, n$)
merge($a, b$)

- If key($a$) < key($b$), change right($a$) to merge(right($a$),$b$)
- Otherwise, change left($b$) to merge($a$,left($b$))
split \((a, n)\)

- If \(cnt = \text{size}(\text{left}(a)) \geq n\),
  Let \(\{l, r\} = \text{split}(\text{left}(a), n)\)
  change \(\text{left}(a)\) to \(r\) and return \(\{l, a\}\)

- Otherwise, Let \(\{l, r\} = \text{split}(\text{right}(a), n - cnt - 1)\)
  change \(\text{right}(a)\) to \(l\) and return \(\{a, r\}\)
Exercise

http://www.spoj.com/problems/MKTHNUM/
http://www.spoj.com/problems/COT/
http://codeforces.com/problemset/problem/191/E
The End