Przemysław Uznański May 22, 2018

Advanced Data Structures

Spring Semester 2018

Exercise Set 10

Cache-Oblivious Lookahead Array: a Cache-Oblivious Data Structure that has the B-Tree functionality, but handles only insertions. In COLA, N elements are partitioned into *disjoint buckets*. Each bucket is just a sorted list of integers. Additionally, each bucket can be of length 2^i only (we say it is on i-th level), and there can be at most one bucket on each level.

Exercise 1:
Show that (predecessor) searching in COLA takes $\mathcal{O}(\log^2(N/B))$ memory transfers. Show that range reporting (reporting all items of value in the given range) takes $\mathcal{O}(\log^2(N/B) + L/B)$ memory transfers, where L is the total number of items reported.

Exercise 2:
Show that insertions into COLA take $\mathcal{O}((\log N)/B)$ *amortized* memory transfers.

Exercise 3:
Show how to augment data stored in COLA so that searching takes only $\mathcal{O}(\log(N/B))$ memory transfers, and range reporting takes $\mathcal{O}(\log(N/B) + L/B)$ memory transfers.

Hint: Fractional cascading: insert some extra values into the arrays so that searching in one helps in searching in next one.

Exercise 4:
So far we analyzed COLA with *growth factor* of 2. Show how to trade insertion time for faster searches, by changing the growth factor to e.g. $g = B^\varepsilon$. Target bound is $\mathcal{O}(\frac{\log(N/B)}{\varepsilon \log B})$ for searching and $\mathcal{O}(\frac{\log N}{B^{1-\varepsilon} \cdot \varepsilon \log B})$ amortized for insertions.

Hint: We still keep at most one bucket per level. This requires us giving a little bit of slack to the bucket sizes: bucket on level i is of size at least g^i and at most $g^{i+1} - 1$.