CS2040S Tutorial 5

AY 24/25 Sem 2 — github/omgeta

Q1. Time complexity of operations on AVL is $O(L \log N)$ vs $O(L)$ for trie. Space complexity for both is $O(L)$ but tries have more overhead cost.

Q2. (a.) For each subtree, there is a horizontal or vertical split. For a horizontal split, check x to split value and go left or right. Vice versa for vertical split on y coord, going up or down. Time complexity: $O(h)$

(b.) Use QuickSelect to select median at each level and partition. Time complexity: $O(n \log n)$

(c.) At horizontal split, recurse on left child. At vertical split, recurse on both children. Time complexity: $O(\sqrt{n})$

Q3. In a Trie, store a count of names under the node, for each gender. On insertion, update counts at the nodes. `countPrefix`: return count stored at node. `countName`: search for node and if end of word, return count of node − count of child nodes. `countBetween`: Find begin and end node ranks, then return $endRank - beginRank - countName(begin)$

Q4. (a.) Using a Trie storing bits, insert numbers from MSB to LSB. To find the best best value of y to XOR with x: if the current bit is 0, go down bit 1 if it exists else 0.