

10 Algorithms (djw1005)

This question concerns unbalanced binary search trees. In some scenarios, most searches are for a small interval in the key space. For example, if we are searching a social media feed for posts keyed by timestamp, perhaps 90% of searches might be for the newest 5% of posts.

- (a) Explain why, in a balanced binary search tree with N items, the worst-case cost of searching for an item is $\Theta(\log N)$. [1 mark]
- (b) Define *amortized cost*. [2 marks]
- (c) Consider an unbalanced binary search tree with N items, where the root holds a dummy key, the left subtree holds $(1 - \alpha)N$ items, the right subtree holds αN items, and where each subtree is balanced. Call this an α -BALANCED tree.

Suppose there are m searches, $(1 - \beta)m$ for items on the left and βm for items on the right. Find the amortized worst-case cost of a search. For $\alpha = 5\%$ and $\beta = 90\%$, would you prefer α -BALANCED or a fully balanced tree? [5 marks]

- (d) Describe the *weighted union* heuristic and the *path compression* heuristic, for the Disjoint Set data structure. Explain what is meant by a *rotation* of a binary search tree. [6 marks]
- (e) We speculate that for real-world search frequencies it might be useful to have an binary search tree that is unbalanced not just at the root but also at other levels. We would also like the data structure to adjust its shape automatically, as searches occur.

By adapting the two heuristics above, suggest a suitable data structure. [*Note:* You do not need to give detailed pseudocode, but you should explain how you are adapting the heuristics.] [6 marks]