

Problem Set 6: Trees

Released 03 April 2024

Due 10 Apr 2024 10:30pm in Gradescope

Instructions. You may discuss the material with your classmates, but you must write solutions yourself. List collaborators on your submission. Do not use other sources of help other than the course staff and the textbook.

If you are asked to design an algorithm, please provide: (a) the pseudocode or precise description in words of the algorithm, (b) an explanation of the intuition for the algorithm, (c) a proof of correctness, (d) the running time of your algorithm and (e) justification for your running time analysis.

Submissions. Please submit a PDF file typed using L^AT_EX. Please assign pages to questions in Gradescope.

1. **12.2-4** Suppose that the search for a key k in a BST ends at a leaf. Consider 3 sets: A the keys to the left of the path, B the keys on the path, and C the keys to the right of the path. Claim: for any $a \in A$, $b \in B$, $c \in C$ $a \leq b \leq c$. Give a smallest possible counterexample
2. **12.3-5** Is deletion commutative. I.e. does deleting x then y always give the same tree as deleting y then x ?
3. **13.1-6** What is the largest number of internal nodes in a red-black tree with black-height k ? The smallest number?
4. **13.1-8** Argue that in a red-black tree a red node cannot have exactly one non-nil child.
5. **13.4-1** Show that if node y in RB-delete is red then no black-heights change.

Bonus (token opportunity). **13.2-5** Right conversion using Right-Rotate