

15-451/651 Algorithm Design & Analysis, Fall 2023

Recitation #4

Objectives

- Familiarize with the SegTree data structure and using it to speed up algorithms
- Understand how to build SegTrees with custom associative operators
- Practice analyzing Splay Trees

Recitation Problems

1. (**Crossing intervals**) Suppose we have a list of n intervals $I_i = [a_i, b_i]$ where $0 \leq a_i, b_i < 2n$, such that the endpoints of all of the intervals are distinct (no two intervals ever share an endpoint at either end). A pair of intervals I_i and I_j are *crossing* if they overlap but one does not strictly contain the other. We want to devise an algorithm to count the number of pairs of crossing intervals.



- (a) Give a simple $O(n^2)$ algorithm for the problem

(b) Come up with a more efficient $O(n \log n)$ algorithm by making use of a SegTree

2. **(Abby's Favorite Problem)** Suppose we start with some array of integers A . Given a sequence of query intervals in the form $[l_1, r_1), [l_2, r_2), \dots, [l_m, r_m)$, return the maximum element and how many times it appears in $A[l_i], \dots, A[r_i - 1]$ in $O(\log n)$ time for each query $[l_i, r_i)$

3. (Splaying Red and Blue)

Consider a splay tree with $n + m$ nodes where n are red and m are blue. Choose weights and use the Access Lemma to prove the following:

(a) The amortized number of splay steps done when a red node is accessed is $4 + 3 \log n$.

(b) The amortized number of splay steps done when a blue node is accessed is $4 + 3 \log m$