

PSO 2 — Week 4: Problems

CS381, Fall 2026

Problem 1. Intervals

You are given n closed intervals $[\ell_1, r_1], \dots, [\ell_n, r_n]$.

- (a) Endpoints are **not** assumed distinct. Find a point contained in the largest number of intervals, together with that number, in $O(n \log n)$ time. State your sorting order including the rule for coordinates where a left endpoint and a right endpoint coincide, prove correctness, and give a concrete two-interval input on which the opposite rule reports the wrong answer.
- (b) Count the pairs $i < j$ whose intervals intersect, in $O(n \log n)$ time. Counting the intersecting pairs directly is awkward; find a quantity that is easier to count and recover the answer from it. Justify that your count has no double counting.
- (c) (*harder*) Assume now that all $2n$ endpoints are distinct. A *nested chain* is a sequence $i_1, \dots, i_t$ with $i_1 \subsetneq i_2 \subsetneq \dots \subsetneq i_t$. Compute the length of the longest nested chain in $O(n \log n)$ time. Prove that the objects your reduction counts correspond exactly to nested chains, in both directions.

Problem 2. Counting with divide and conquer

Let $A[1..n]$ be an array of n distinct integers in arbitrary order. The values may be enormous, much larger than n , and are not bounded by any polynomial in n . Say where this assumption constrains your algorithms.

- (a) An *inversion* is a pair $i < j$ with $A[i] > A[j]$. Count the inversions of A in $O(n \log n)$ time.
- (b) A *significant inversion* is a pair $i < j$ with $A[i] > 3A[j]$ (assume all entries are positive). Count them in $O(n \log n)$ time. Explain why you cannot count these during the merge step the way you did in (a).
- (c) (*harder*) Fix an integer $D \geq 1$. Count the *D-local* inversions: pairs $i < j$ with $A[i] > A[j]$ and $j - i \leq D$. Give an $O(n \log n)$ algorithm, with the bound independent of D . Then explain why the divide-and-conquer scheme of (a) does not adapt, and why the natural repair, cutting the array into blocks of length D and running (a) on each pair of adjacent blocks, gives the wrong count.