Week 4 PSO Questions

TAs

Mukul Agarwal Aaryan Wadhwani

September 15, 2026

Second Smallest Distance.[1] Recall that given a set P of n points in $\mathbb{R}^2$, we can compute the minimum distance between any pair of points in P in $O(n \log n)$ time. Consider now the problem of computing the second smallest distance between any pair of points in P. Design and analyze an algorithm for this problem.

Six-Coloring a Planar Graph.[2] Design and analyze a linear time algorithm for six-coloring a planar graph.

Schedule-slopping Friendslop.[3] Recently, *Big Walk*, a new hit Friendslop game, released to critically acclaimed reviews. Justin would like to play *Big Walk* with his friends, but they are all old adults with jobs and prior commitments or something. Your objective is to find the number of times which Justin and his friends can all play (assuming there is such a time). They only want to play *Big Walk* if they can all meet at least 6 times (or 7 if they're bad).

Formally, there are m friends. Each friend i has a list of times they are busy represented by disjoint, sorted ranges $L_i = \{[a_1^i, b_1^i], ..., [a_t^i, b_t^i]\}$. Each range $[a_j^i, b_j^i]$ represents the *inclusive* start a_j^i and end b_j^i of their busy times. Assume time (measured in some arbitrary unit) starts at 0 and ends at the largest number given in the input. In total, there are n busy times. Give an algorithm that takes $(L_1, L_2, ..., L_m)$ as input and outputs the number of times they can meet. In other words, output the number of occurrences where there is a gap between busy times.

Example ($m = 2, n = 5$):

[1]This problem is due to Kent Quanrud.
[2]This problem is due to Kent Quanrud.
[3]This problem is due to Justin Zhang.

Input: $L_1 = [(1, 2), (4, 6), (8, 12)]$
$L_2 = [(1, 3), (5, 9)]$

Output: 2 because the free times are $[0, 1)$ and $(3, 4)$.