COS 330: Great Ideas in Theoretical Computer Science

Fall 2026

Problem Set 7

Module: New Computational Models

Below is a reminder of key aspects of the PSet:

- **The only goal of this PSet is to help you develop your problem-solving skills.** Your performance on this PSet will not directly contribute to your grade, but will indirectly improve your ability to do well on the exams.
- **Your performance does not directly impact your grade**, so you may use any resources you like (collaboration, AI, etc.) to help you complete the PSet.
- **This PSet has a single problem, and it is exam-style.** We recommend attempting it alone first, before using AI or other help.
- **None of this is an integrity rule** – PSets are not graded directly. It's advice about what will actually develop your skills and in the process prepare you well for the exams, which are closed-book and closed-AI.
- **See the course website** for a full walkthrough of AI tools and recommendations.
- **An AI-helper is available for Problem 1** – unlike a general assistant, it gives hints and pushes back instead of handing you the answer. See the link above for more information.

Aligning Expectations

The 🗎 symbol marks an exam-style problem. We highly recommend writing up a full solution as if you were on an exam.

Problem 1 : Grouping Points on a Line 🗎

A sequence of (not necessarily distinct) integer points $x_1, \ldots, x_n \in \mathbb{Z}$, where $n \geq 1$, is given online, one point at a time. Given a length $L > 0$, we want to partition the points into as few groups as possible, with the leftmost and rightmost points in each group at most L apart.

We represent a group by a nonempty set $S \subseteq \{1, \ldots, n\}$ of indices. Its span is

$$I(S) = \left[\min_{i \in S} x_i,\ \max_{i \in S} x_i\right].$$

The length of a group is the length of its span:

$$\text{len}(S) = \max_{i \in S} x_i - \min_{i \in S} x_i.$$

A group is valid if $\text{len}(S) \leq L$. A valid solution is a partition of $\{1, \ldots, n\}$ into valid groups. Its cost is the number of groups in the partition.

The points arrive one at a time, in the order $x_1, \ldots, x_n$. When x_i arrives, the algorithm must add it to an existing group or start a new group, **before seeing the next point**. Assignments are permanent: **points cannot be moved between groups**, and **groups cannot be merged**. Adding a point may extend a group's span, but **its length must stay at most** L.

Let OPT denote the minimum number of valid groups needed when the entire sequence is known in advance. This is the optimal offline cost.

Consider the following **greedy algorithm**. When a point x_i arrives:

1. If x_i lies in an existing group's span, including its endpoints, assign it to that group.
2. Otherwise, if adding x_i to an existing group would keep its span length at most L, assign it to such a group.
3. If neither case applies, create a new group containing just x_i.

Break ties arbitrarily.

Let G be the number of groups returned by greedy. We will analyze the competitive ratio of this algorithm.

(a) Let $L = 2$. For the arrival sequence $0, 2, -1, 3$, determine the groups returned by greedy and the value of OPT. Justify your answers.

(b) Prove that after each arrival, the greedy groups satisfy both of the following properties:
(i) The union of any two distinct groups is not a valid group.
(ii) The spans of distinct groups are disjoint.

(c) Prove that the greedy algorithm is 2-competitive, in other words, prove $G \leq 2\text{OPT}$ for every input and every choice of ties in the greedy algorithm.

Small Hint

Try to select as many input points as possible that are pairwise more than L apart. These points certify that OPT is at least the number of selected points, since no valid group can contain two of them.

(d) For every integer $k \geq 1$, construct an input with $\text{OPT} = k$ for which greedy returns $2k - 1$ groups. Prove both claims. You may use the example in part (a) as a starting point.

To clarify, for each given k, specify a value of L and a finite sequence of integer points, including their arrival order. Show that greedy creates exactly $2k - 1$ groups, and that a valid partition into k groups exists but no valid partition uses fewer than k groups.