Collaboration Policy: For this homework, Problems 1–2 can be worked in groups of up to three students. Submission is online on moodle.

1. (50 pts.) Is your overlap in vain? (Dynamic programming.)

Let \( \mathcal{I} \) be a set of \( n \) closed intervals on the real line (assume they all have distinct endpoints). A set of intervals \( \mathcal{X} \subseteq \mathcal{I} \) is **admissible** if no point on the real line is covered by more than 3 intervals of \( \mathcal{X} \). Let \( C(\mathcal{X}) \) be the the set of all points on the real line that are covered by two or more intervals of \( \mathcal{X} \). The **profit** of \( \mathcal{X} \), denoted by \( \phi(\mathcal{X}) \), is total length of \( C(\mathcal{X}) \).

Describe an algorithm, as fast as possible, that outputs the subset of \( \mathcal{I} \) that maximizes the profit, and is admissible.

(Hint: Look on the class slides for dynamic programming.)

\[
\begin{array}{c}
\text{interval 1} \\
\text{interval 2} \\
\text{interval 3} \\
\text{interval 4} \\
\text{interval 5} \\
\end{array}
\Rightarrow
\begin{array}{c}
\text{interval 1} \\
\text{interval 2} \\
\text{interval 3} \\
\text{interval 4} \\
\text{interval 5} \\
\end{array}
\]

\[C(\mathcal{X}) = 3 + 3 + 5 = 11.\]

2. (50 pts.) Diagonals and points. (Dynamic programming + DAGs and topological sort.\(^1\))

(A) (10 pts.) The **rank** of a vertex \( v \) in a DAG \( G \), is the length of the longest path in DAG that starts in \( v \). Describe a linear time algorithm (in the number of edges and vertices of \( G \)) that computes for all the vertices in \( G \) their rank.

(B) (10 pts.) Prove that if two vertices \( u, v \in V(G) \) have the same rank (again, \( G \) is a DAG), then the edges \((u, v)\) and \((v, u)\) are not in \( G \).

(C) (10 pts.) Using (B), prove that in any DAG \( G \) with \( n \) vertices, for any \( k \), either there is a path of length \( k \), or there is a set \( X \) of \( \lfloor n/k \rfloor \) vertices in \( G \) that is independent; that is, there is no edge between any pair of vertices of \( X \).

(D) (10 pts.) Consider a set of \( P \) of \( n \) points in the plane. The points of \( P \) are in general position – no two points have the same \( x \) or \( y \) coordinates. Consider a sequence \( S \) of points \( p_1, p_2, \ldots, p_k \) of \( P \), where \( p_i = (x_i, y_i) \), for \( i = 1, \ldots, k \). The sequence \( S \) is **diagonal**, if either

- for all \( i = 1, \ldots, k - 1 \), we have \( x_i < x_{i+1} \) and \( y_i < y_{i+1} \), or
- for all \( i = 1, \ldots, k - 1 \), we have \( x_i < x_{i+1} \) and \( y_i > y_{i+1} \).

Prove using (C) that there is always a diagonal of length \( \lfloor \sqrt{n} \rfloor \) in \( P \). Describe an algorithm, as fast as possible, that computes the longest diagonal in \( P \).

(E) (10 pts.) Using the algorithm of (D), describe a polynomial time algorithm that decomposes \( P \) into a set of \( O(\sqrt{n}) \) disjoint diagonals. Prove the correctness of your algorithm.

---

\(^1\)If you do not know what topological sort is, and how to compute it in linear time, then you need read on this stuff – you are suppose to know this, and you might be tested on this stuff.