CS 473 (Fall 2026)
Homework 9 (due Nov 19 Thu 10am)

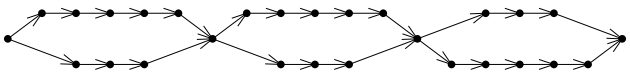
Instructions: Carefully read `http://courses.grainger.illinois.edu/cs473/fa2026/policies.html#hw` and `http://courses.grainger.illinois.edu/cs473/fa2026/integrity.html` (in particular, remember to include the 3 disclosure statements, on collaborators, outside sources, and AI use, for each problem).

Problem 9.1: Consider the following problem Colorful-Path:

> *Input*: a directed acyclic graph (dag) $G = (V, E)$ with n vertices and m edges, and vertices $s, t \in V$, and a color $c(e) \in \{1, \ldots, M\}$ for each edge $e \in E$, and a number ℓ.
>
> *Output*: yes iff there exists a path from s to t that encounters at least ℓ *distinct* colors.

Prove that Colorful-Path is NP-hard, by giving a polynomial-time reduction from 3-SAT to Colorful-Path. Remember to describe precisely your construction of an input to Colorful-Path from any given input to 3-SAT (a 3CNF formula – note that we are not given a satisfying assignment itself!), and carefully prove correctness of your reduction.

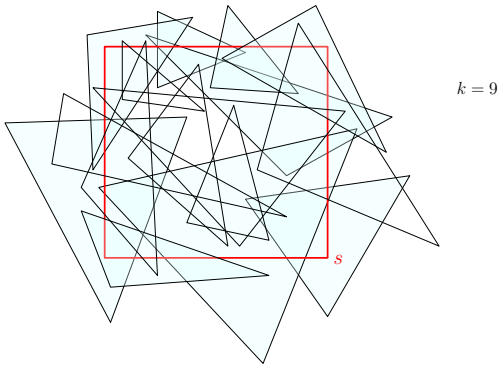
(Hint: for your construction of the dag G, try something like the picture below... Describe how to assign colors to the edges along these paths.)



Problem 9.2: Consider the following problem COVERING-SQUARE-BY-TRIANGLES:

> *Input*: a set T of n triangles in 2D, a square s, and an integer k.
>
> *Output*: yes iff there exists a subset $Q \subseteq T$ of k triangles such that the union of Q covers the entire boundary of s.

(Note: we don't require the union of Q to cover the interior of s. Also, the variant of the problem of covering just one side of s, rather than the entire boundary of s, is easier and can actually be solved in polynomial time by a greedy algorithm.)



Prove that COVERING-SQUARE-BY-TRIANGLES is NP-hard.

You may use the fact that VERTEX-COVER is NP-hard even when restricted to graphs with maximum degree 3. You may also use the fact that any graph with maximum degree d is $(d+1)$-edge-colorable, i.e., we can assign a color from $\{1, \ldots, d+1\}$ to each edge, so that no two adjacent edges have the same color; such a coloring can be found in polynomial time.

As always, describe the construction of your reduction precisely, and carefully prove correctness of your reduction.

(Hint: map each edge of the input graph to a tiny interval along the boundary of s...)