Homework 3

Due: October 4, 2022 at 14:30 ET

This homework must be typed in LaTeX and handed in via Gradescope.

Please ensure that your solutions are complete, concise, and communicated clearly. Use full sentences and plan your presentation before you write. Except in the rare cases where it is indicated otherwise, consider every problem as asking you to prove your result.

Problem 1

1. Let E be the set of edges of a finite undirected graph and ℓ be the collection of subsets of E such that $A \in \ell$ if and only if A is an acyclic subset of E. Show that the ordered pair (E, ℓ) is a matroid.

2. Let $w : E \to \mathbb{R}^+$ be a function that assigns a non-negative weight to each element of E. Given a subroutine that takes as inputs an edge (u, v) in E and an acyclic subset A of E, and returns *true* if $A \cup (u, v)$ is acyclic and $false$ otherwise, design an efficient algorithm to find an acyclic subset of E of maximum weight. Analyze its runtime and argue the correctness/optimality.

Problem 2

A *subsequence* of $[a_1, a_2, \cdots, a_n]$ is a list of numbers $[a_{i_1}, a_{i_2}, \cdots, a_{i_k}]$ such that $i_1 < i_2 < \cdots < i_k$.

Consider the list $[5, 1, 6, 2, 3, 10, 11]$. Two strictly increasing subsequences are given by $[5, 6, 10]$ and $[1, 2, 3, 10, 11]$. Moreover, $[1, 2, 3, 10, 11]$ is the longest strictly increasing subsequence of this list of numbers.

1. Given an arbitrary list of numbers, design a greedy algorithm to determine the length of the longest strictly increasing subsequence. Your algorithm should run in $O(n \log n)$ time where n is the length of the input list. Analyze the runtime of your algorithm and argue correctness/optimality.

2. Modify your algorithm to instead output the longest increasing subsequence of the original list of numbers. In the case of multiple longest increasing subsequences, outputting any of them shall suffice. Argue the correctness of your modified algorithm.