

# ECE 374 B ✧ Spring 2024

## 🌀 Homework 7 🌀

---

- **Submit your solutions electronically on the course Gradescope site as PDF files.** If you plan to typeset your solutions, please use the  $\text{\LaTeX}$  solution template on the course web site. If you must submit scanned handwritten solutions, please use a black pen on blank white paper and a high-quality scanner app (or an actual scanner, not just a phone camera).
- 

### 👉 Some important course policies 👉

- **You may use any source at your disposal**—paper, electronic, or human—but you *must* cite *every* source that you use, and you *must* write everything yourself in your own words. See the academic integrity policies on the course web site for more details.
  - **Avoid the Three Deadly Sins!** Any homework or exam solution that breaks any of the following rules will be given an *automatic zero*, unless the solution is otherwise perfect. Yes, we really mean it. We're not trying to be scary or petty (Honest!), but we do want to break a few common bad habits that seriously impede mastery of the course material.
    - Always give complete solutions, not just examples.
    - Always declare all your variables, in English. In particular, always describe the specific problem your algorithm is supposed to solve.
    - Never use weak induction.
- 

### See the course web site for more information.

If you have any questions about these policies,  
please don't hesitate to ask in class, in office hours, or on Piazza.

---

1. You are given a list  $D[n]$  of  $n$  words each of length  $k$  over an alphabet  $\Sigma$  in a language you don't know, although you are told that words are sorted in lexicographic order. Using  $D[n]$ , describe an algorithm to efficiently identify the order of the symbols in  $\Sigma$ . For example, given the alphabet  $\Sigma = \{Q, X, Z\}$  and the list  $D = \{QQZ, QZZ, XQZ, XQX, XXX\}$ , your algorithm should return  $QZX$ . You may assume  $D$  always contains enough information to completely determine the order of the symbols. (Hint: use a graph structure, where each node represents one letter.)
  
2. Given a directed-acyclic-graph ( $G = (V, E)$ ) with integer (positive or negative) edge weights:
  - (a) Give an algorithm to find the **shortest** path from a node  $s$  to a node  $t$ .
  - (b) Give an algorithm to find the **longest** path from a node  $s$  to a node  $t$ .
  
3. You are given a **directed-acyclic-graph**  $G = (V, E)$  with possibly negative weighted edges:
  - (a) Give an algorithm that finds the shortest path that contains at most  $k$  edges between two vertices  $u$  and  $v$  in  $O(k(n + m))$  time.
  - (b) Give an algorithm that finds the shortest path that contains exactly  $k$  edges between two vertices  $u$  and  $v$  in  $O(k(n + m))$  time.

*Hint: You can solve both problems almost the same way. Modify the graph  $G$  and utilize the algorithm from problem 2 part (a).*