You are allowed to discuss with others but are not allowed to use any references other than the course notes and the reference books. Please list your collaborators for each question. You must write your own solutions. See the course outline for the homework policy.

This homework accounts for 8% of your total grade.

For all problems below, you can assume that you are working under the word RAM model.
Unless otherwise stated, giving an algorithm means providing a high level description, pseudocode (if needed to clarify the algorithm), a correctness argument, and runtime analysis.

Due date: November 6th at 11:59pm

PROBLEM 1 (20 Points) - Number of shortest $s - t$ paths

Given an undirected graph $G = (V, E)$ and two vertices s and t, design an algorithm to return the number of different shortest $s-t$ paths between s and t. Two paths are the same if they have exactly the same sequence of vertices; otherwise they are different. You will get full marks if the time complexity of the algorithm is $O(|V| + |E|)$ word operations and the proofs are correct. We assume that the answer fits in one word, so that adding two numbers with the sum being at most the answer can be done in one word operation.

PROBLEM 2 (20 Points) - Directed Odd Cycle

Given a directed graph $G = (V, E)$, design an algorithm to determine if there is a directed odd cycle in G or not. You will get full marks if the time complexity is $O(|V| + |E|)$ and the proofs are correct.

Problem 3 (20 Points) - Escaping a maze with one reversal

Let $G = (V, E)$ be a **directed** graph with n vertices and m edges. Suppose the vertices are labeled $\{1, 2, ..., n\}$, and edges are provided as an **array** E **of** (s, t) **pairs**, where the edge (s, t) goes from vertex s to vertex t.

1. [2 marks] Provide pseudocode for a procedure *GetAdjRepr* that takes n, m and the array of edges as input, and returns an adjacency list representation A of the graph. In this part, you do **not** need to give a high level description, or correctness argument.

2. [2 marks] Provide pseudocode for a procedure *ReverseAdj* that takes n, m and **the output of (a)** as input, and returns an adjacency list representation R of the *reverse graph* in which the direction of every edge is reversed (so if $(s, t) \in A$ then $(t, s) \in R$). In this part, you do **not** need to give a high level description, or correctness argument. **Note: arguments are NOT the same as in (a).**

3. [16 marks] Suppose you are in a maze and you want to escape. The vertices of G are the locations of the maze, and each edge (s, t) describes a movement you can make: specifically from location s to location t. Your goal is to move from where you are, at location 1, to the exit at location n. You have access to a powerful device that can reverse the direction of all edges, allowing you to move backwards along edges. You can use this device whenever you want, but you can only use it once, and its effects are permanent. Once you use it, all edges remain "reversed" forever, and you can only travel along them in their "reversed" direction.

   Give an algorithm that determines whether you can escape the maze, returning true if so, and false otherwise.

PROBLEM 4 (20 Points) - Tracking a wildfire

A farmer has fields that are modeled as squares on $n \times n$ two-dimensional grid. The grid of cells has n rows going from north to south and numbered from 1 to n, and n columns going from west to east numbered from 1 to n. So coordinates of north-west corner are $(1, 1)$, and coordinates of south-east corner are (n, n). Each grid cell either contains corn or water. This data is represented as array F of characters, with F_{ij} equal to "c" or "w" (representing "corn" or "water" respectively).

A fire starts in a corn cell $(x, y), 1 \leq x, y \leq n$ at time 0, and simultaneously spreads one cell in each direction (North, South, East, West) in each unit of time (unless the cell was already burned or contains water). It also take one unit of time for the cell of corn to be completely burned. The fire cannot spread to water (and thus can be blocked by water).

Describe a graph representation of this problem and answer the following question by providing an algorithm (based on standard algorithms considered in class) and run-time analysis in terms of n. In particular, if you build a graph with n^2 vertices ($V = \{1, 2, \ldots, n^2\}$) give an explicit construction of set of edges E.

1. [5 marks] Provide a pseudocode description of an algorithm CONVERTTOGRAPH that takes F, and n as parameters and returns graph (a pair (V, E)) in your favourite representation.
2. [5 marks] will all corn burn?
   Provide a pseudocode description of an algorithm WILLITBURN that takes F and coordinates x, y as parameters and returns TRUE or FALSE. Correctness can be brief.
3. [5 marks] how long will it take for the fire to be over?
   Provide a pseudocode description of an algorithm TIMETOBURN that takes F and a coordinates x, y as parameters and returns integer value which represents the time the fire is over. Correctness can be brief.
4. [5 marks] if not all corn will burn, how many cells with corn will remain?
   Provide a pseudocode description of an algorithm SAVEDFIELDS that takes F and a coordinates x, y as parameters and returns the number of cells left unharmed. Correctness can be brief.

PROBLEM 5 (20 Points) - Bottleneck spanning trees

Consider a connected graph $G = (V, E)$ with n vertices and positive edge weights $w_e > 0$ on the edges $e \in E$. Assume that the w_e's are *distinct* in G. Let $T = (V, E')$ be a spanning tree of G. The *bottleneck edge* of T is the edge in E' with the highest edge weight. A spanning tree T_b of G is a *minimum bottleneck spanning* (MBST) tree of G is there is no other spanning tree T' of G with a smaller bottleneck.

1. [14 marks] Prove that every minimum spanning tree (MST) of G is also an MBST.
2. [6 marks] Give a counterexample to the claim that an MBST is always an MST.