Problem 1

Given are $n$ courses and for each course given are its prerequisites. Let $P_i$ be the set of prerequisite courses for the $i$-th course and let $m = |P_1| + |P_2| + \ldots + |P_n|$. Give an $O(n + m)$ algorithm that finds the size of the longest prerequisite chain, i.e., the longest sequence of courses for which for every element in the sequence the previous element is its prerequisite. You may assume that the data is consistent, i.e., there are no “prerequisite loops.”

Problem 2

In a town full of one-way streets, the town council suddenly realized that it is not possible to get from every location to every other location. They want to fix this but their budget is limited – they can build a single new road. Help them!

Given is a directed graph $G = (V, E)$ (where vertices represent crossings and edges one-way streets). Determine if there exists a pair of vertices $u$ and $v$ such that adding an edge from $u$ to $v$ makes the graph strongly connected. Your algorithm should run in time $O(n + m)$, where $n = |V|$ and $m = |E|$.

Problem 3

Given is an $m \times n$ board where every position corresponds to either an empty space or a wall. There are also two little robots, each occupying an empty space. In a single move, each robot can move along a direction parallel to the axes of the board until it hits a wall or the other robot (it stops on the empty space just before the wall or the other robot). The robots cannot stop midway and it is really bad if either of them falls off the board. Given the robots’ starting positions and a pair of target locations that the robots need to get to, find the smallest number of moves the robots need to make to get to the target locations, or determine that it is not possible to get there. It does not matter which robot gets to which target location. A robot can get to a target location and then leave it to help the other robot get to their target location. At the end of the sequence of moves, both target locations need to be occupied by a robot. Your algorithm should run in time $O((mn)^2)$. 