3rd tutorial

1. Pouring Water. We have three containers whose sizes are 10 pints, 7 pints, and 4 pints, respectively. The 7-pint and 4-pint containers start out full of water, but the 10-pint container is initially empty. We are allowed one type of operation: pouring the contents of one container into another, stopping only when the source container is empty or the destination container is full. We want to know if there is a sequence of pourings that leaves exactly 2 pints in the 7- or 4-pint container.

(1) Model this as a graph problem: give a precise definition of the graph involved and state the specific question about this graph that needs to be answered.
(2) What algorithm should be applied to solve the problem?
(3) Find the answer by applying the algorithm.

2. Unique. How to recognize graphs which can be topologically ordered in exactly one way?

3. #Paths in a DAG. Given a DAG G and its two vertices u, v, design an algorithm computing the number of distinct paths from u to v.

4. #Shortest Paths. Given a directed graph G and its two vertices u, v, design an algorithm computing the number of shortest paths from u to v.

5a. Parallel Planning. Imagine you are building a house or starting a business, and you have a *dependency graph*. This is a DAG $G = (V, E)$ whose vertices are tasks and an edge (u, v) means that task u has to be finished before task v can be started. Moreover, you have a length function $\ell : V \to \mathbb{N}$ which says that task u takes $\ell(u)$ time units to complete.
You are given a target date $T \in \mathbb{N}$, and you need to compute, for each task $u \in V$, what is the latest time $t(u)$ at which you can start executing this task so that everything is finished by time T. Assume you are able to do arbitrarily many tasks in parallel.

5b. Critical Vertices. Call a vertex u in G defined above *critical* if it corresponds to a task which, if it becomes delayed (equivalently, if $\ell(u)$ increases), the time the last task finishes also increases (i.e., the whole project is delayed). How to detect all critical vertices?

---

6. Broken Taxi. A taxi broke down in Manhattan, so it can only drive straight or turn right. (Manhattan is an $n \times n$ grid.) You are given the location of the taxi, the location of a repair shop, and your task is to give instructions to the driver on how to reach the shop and burn as little gas as possible.

7. Ghost. There is a maze and at position S a ghost. Think of the maze as a grid where each cell is either a hallway or a wall. The ghost can walk through walls, but it moves through walls $4\times$ slower than through hallways. The ghost wants to reach a position T. Find the shortest path.

8. Guard. We have a maze on an $N \times N$ grid with walls, a hero, and a treasure. We want to find a way for the hero to reach the treasure as quickly as possible, but not run into a guard (that is, appear in the same cell as the guard at the same time). The guard has a fixed route of length L as a sequence of adjacent cells and it goes back and forth on this route. How to find such a shortest route as quickly as possible? What is the complexity of your algorithm in terms of N and L?

9. Lost Robots. There is a maze and 2 robots at different places in the maze. However, these robots are controlled by the same remote control, which has four buttons – going north, south, east, and west. When a robot receives a command which cannot be executed, it ignores it. How to find a sequence of commands which takes both robots out of the maze? (Once a robot is out of the maze, it stops listening to commands.) How to find the shortest such sequence?

10. Silly Robot. Again a grid maze with hallways and walls. In it, a robot which goes straight and only turns if it hits a wall or the boundary of the maze (i.e., reaches a cell adjacent to a wall, and reaches this cell from a direction orthogonal to the wall). You select the direction in which it turns (left, right, backwards). Find a path from a start cell s to a target cell t with the least number of turns.

11. Longest Common Subsequence. We have two sequences (e.g. long numbers, words, sentences) and we want to find their longest common subsequence. (A subsequence consists of characters which are not required to be consecutive.) How to do it (as a graph problem)?