1. Three water containers have capacities of 10, 7 and 4 liters, respectively. Initially the 7 and 4 liter containers are full and the 10 liter container empty. The task is to measure exactly 2 liters into any one of containers. The only allowed operation is pouring water from one container into another until either the first container is empty or the second one is full.

(a) Model the task using graphs. What precisely is the graph, and what is the precise question we are asking about the graph?

(b) Sketch an algorithm for the problem. How would you implement the graph? If you use some of the graph algorithms from the course, do they need to be somehow modified? Do you need some additional data structures? Estimate the time and memory requirement of your algorithm in the general case, where the contained capacities can be arbitrary positive integers.

You need not produce a “best possible” algorithm. Just develop some working basic solution and then think a little about possible ideas for improvement. This kind of problems are discussed more extensively on the course *Artificial Intelligence*.

2. A graph $G = (V, E)$ is given as an adjacency list. Describe roughly an algorithm for creating the adjacency list representation for the component graph $G_{SCC} = (V_{SCC}, E_{SCC})$.

Naturally, you may use the algorithm *STORONGLY-CONNECTED-COMPONENTS* as a part of your solution. You do not need to give precise pseudocode, but explain what intermediate results you compute and what data structures you need. Take particular care to avoid creating duplicates into the adjacency lists of $G_{SCC}$.

3. Find the strongly connected components of the graph below using the algorithm from the class. Assume that the adjacency list of each node is in alphabetical order.

What is the component graph here? What is the least number of edges you would need to add to the original graph to make it strongly connected?

See next page!
4. Find the shortest paths from node $s$ in the graph below. Use the Bellman-Ford algorithm. Assume that the adjacency list of each node is in alphabetical order.