Theory Questions
1. What is an algorithm? A sequence of steps such as a recipe or a movie script.

As a movie viewer, please write an algorithm for a process to "see a movie". Include details of the following for example (add additional information yourself):

- The start conditions
-- I want to watch a movie
- Roles (people/actors involved)
--watcher, ticket seller etc
-Equipment required ("props")
--seats, ticket etc
- The scenes
-- scene 1 - buy ticket
-- ??? what happens next
-- scene 3, scene 3 ...
- Results/outcomes
-- ?
your answer here...

Screenplay:
(1.) the conditions of play the movie:
(2.) (a)I want to see a movie
(3.) (b)I have enough money to buy a movie ticket
(4.) (2.) role:
I,the conductor,the projector,the doorman
(5.) (3.) props: Movie tickets,chairs,projection equipment(screens, video cameras), money
(6.) (4.)the scene: Scene one: buy a ticket
(7.) (a)I walked to the ticket office and take out the money to the conductor.
(8.) (b)the conductor took the money and gave me a movie ticket.
Scene two: go to the cinema
(a)I took the ticket, walked into the entrance, and took out the ticket for the doorman.
(b)the doorman let me in.
Scene three: waiting for the movie to start, I find my seat and sit down. Scene four: see a movie
(a)the movie started
(b)I was deeply attracted by the story and absorbed in the movie.
Scene five: The film is finished
(a)the film is over
(b)I left the cinema with others.

(9.)
(10.) (5.)results (a.)I had a good mood (b)I spent the money
(11.) (C.)the cinema earned money.
(12.) ________________________________
(13.) 2. A farmer with his wolf, duck and bag of corn come to the east side of a river they wish to cross. There is a boat at the rivers edge, but of course only the farmer can row. The boat can only hold two things (including the rower) at any one time. If the wolf is ever left alone with the duck, the wolf will eat it. Similarly if the duck is ever left alone with the corn, the duck will eat it. How can the farmer get across the river so that all four arrive safely on the other side?
(14.) Initial State
(15.) (1) Farmer takes duck to left bank
(16.) (2) Farmer returns alone to right bank
(17.) (3) Farmer takes wolf to left bank
(18.) (4) Farmer returns with duck
(19.) (5) Farmer takes corn to left bank
(20.) (6) Farmer returns alone
(21.) (7) Farmer takes duck to left bank
(22.) (8) Success
(23.)

3.Explain why we determine our problem goals before we write the problem formulation (including the model and deciding what algorithm or techniques to use - for example search, or other method).

(24.) See Chapter3.1 (Russel & Norvig)
(25.) In goal formulation, we decide which aspects of the world we are interested in, and which can be ignored or abstracted away. Then in problem formulation we decide how to manipulate the important aspects (and ignore the others). If we did problem formulation first we would not know what to include and what to leave out. That said, it can happen that there is a cycle of iterations between goal formulation, problem formulation, and problem solving until one arrives at a sufficiently useful and efficient solution.
(26.)
(27.) 4. This question requires you to perform BFS and DFS on paper.
(28.)
(29.) Simulate (on pencil-and-paper) breadth-first search starting from node A when the goal node is K.
(30.) Simulate (on pencil-and-paper) depth first search starting from node A when the goal node is I.
(31.) 1.BFS(there are used right hand rule .Maybe you can ues the left hand rule). A:[B] B:[C,D] C:[D,E,F] D:[E,F,H,G] E:[F,H,G] F:[H,G] H:[G,K,I] G:[K,I] K=goal

2. DFS(there are used right hand rule .Maybe you can ues the left hand rule). A:[B] B:[C,D] C:[E,G,D] E:[F,D] F:[D] D:[H,G] H:[I,K,G] I=goal

5.Consider a state space where the start state is number 1 and each state k has two successors: numbers 2k and 2k + 1.

a. Draw the portion of the state space for states 1 to 15.

b. Suppose the goal state is 11. List the order in which nodes will be visited for breadthfirst search, depth-limited search with limit 3, and iterative deepening search.

c. How well would bidirectional search work on this problem? What is the branching factor in each direction of the bidirectional search?

(32.) d. Does the answer to (c) suggest a reformulation of the problem that would allow you to solve the problem of getting from state 1 to a given goal state with almost no search?

(33.) e. Call the action going from k to 2k Left, and the action going to 2k + 1 Right. Can you find an algorithm that outputs the solution to this problem without any search at all?

(34.) See textbook Chapter3.15

(35.) 1.

(36.) Breadth-first: 1 2 3 4 5 6 7 8 9 10 11. Depth-limited: 1 2 4 8 9 5 10 11. Iterative deepening: 1; 1 2 3; 1 2 4 5 3 6 7; 1 2 4 8 9 5 10 11

(37.) Bidirectional search is very useful, because the only successor of n in the reverse direction is $\lfloor (n/2) \rfloor$. This helps focus the search. The branching factor is 2 in the forward direction; 1 in the reverse direction.

(38.) Yes; start at the goal, and apply the single reverse successor action until you reach 1.

(39.) The solution can be read off the binary numeral for the goal number. Write the goal number in binary. Since we can only reach positive integers, this binary expansion beings with a 1. From most- to least- significant bit, skipping the initial 1, go Left to the node 2n if this bit is 0 and go Right to node 2n + 1 if it is 1. For example, suppose the goal is 11, which is 1011 in binary. The solution is therefore Left, Right, Right.

(40.)