1. a) Shopping for used AI books on the Internet

Performance Measure: The measure should be based on the total number of identified used AI books(more the better), the time taken for identifying the used AI books(lesser the better) , the number of good author and well-written books (more the better) and the cost of the book(lesser the better), with the availability of shipment or availability of product at a closer location to the user.

Environment: User, Web platform, other websites

Actuators: Actuators answer the question using what it delivers output ? Hence the internet search for book can be done using the desktop pc for display, giving list of url's to be visited to purchase the book, could email the list of books.

Sensors: Sensors answer the question using what you got the input data? Hence the sensors will be keyboard, the name of book we enter in their search box(the search form)

A state in this problem refers to the books identified by the search agent. Since we cannot decide the book it would output based on the previous or current outputs of the search , it is non-deterministic. Since the environment remains the same irrespective of the number of searches, it is static. The task is observable as it has all relevant information on which it should base its search. This search process is sequential as it uses the search results it got for every variation that the user gives and filters or makes additional search. Hence it uses previous memory due to which for a given search its sequential(that is for AI book). When the search is changed to physic or some other book it becomes episodic. Hence it is episodic between searches of different subjects. Its slightly dynamic as the user will modify the input he gives for the search based on the results displayed.

This system is single agent as there is only one operator.

Its continuous as the search is done on a continuous basis till it displays the results.

b)knitting sweater

Performance Measure: How closely the sweater is knit, its quality, time taken for weaving it

Environment: the weaver,chair,table

Actuator: Hands,needle,woollen thread are used to deliver the net output of sweater.

Sensor: The eyes are responsible for deciding where each weave is to be done and hence the actuator.

A state in this process is every weave. Since we can predict based on the current and previous weave, where next one should be done, it is deterministic. Since the environment doesn't change its static. Its observable as it has all relevant information. The weaving process is sequential as it uses past memory for deciding the next weave. Its single agent.

The entire task is sequential.

2.a) Initial State: To input all the records.

Goal Test State: The record under processing is Illegal or invalid.

State: Each state is processing a record

Transition: To consider consecutive record when the processing of current is over.

Successor function: This results in the state of processing the next record given the current record is not illegal.

Cost function: This would involve the computation time to process each record, the lesser number of records processed before illegal record is identified (each record processed can be taken as unit step value) .

b) Initial state: All three jugs are empty

State: Each state is represented by the total quantity of water measured by all three jugs.

Transition: Poring the contents of a jar onto ground or another jar.

Successor Function : Returns a list of states representing different levels of water in each jar and the actions would represent each pour. An example of the successor {(a,b,c),{$x_a \rightarrow x_b$)} where a,b,c represent the quantity of water in each of the jars and the action is pouring x_a quantity from a jar to b jar which results in x_b quantity.

Goal Test State: When the total quantity summed over three jars equals 1 gallon

Cost Function: The cost would be the number of pouring. Each pouring will be given same value irrespective of the quantity.

3. a).Yes. Breadth first search is a special case of uniform cost search. Basically uniform cost search is done for each level starting from the root and with a priority queue as reference. First it branches out to the first level after the root and calibrates the cumulative cost for each path. The one with least cost and least number of alphabets in case of equal costs is considered priority in the next iteration. This priority path will further be expanded to its children in the next iteration. And then it deduces that path which reaches the goal with least cost. The Breadth first search is similar and becomes a special case of this when all the costs are equal. That is there is no necessity for priority queue and all the quantities in a particular level are given equal priority without any bias.

b) Yes. Depth first search is a special case of Best First search. Best First search basically involves a heuristic function that involves efficient method of identifying the node to be expanded. DFS becomes Best first search when f(n)=-depth(n). That is when the path expanded is selected by expanding the child every time BY using a particular heuristic.

c)Yes. Uniform cost search is a special case of A* search when the heuristic function of A* is zero.