AI HW1

1a)

P: Speed, responding to the guests at the table at the corret time, and handing out the correct number of cards.

E: The poker table, the deck of cards, the guests, casino, chips

A: Arms to deal the cards with and to count the number of cards, a mechanism for which to shuffle the cards.

S: Card count sensor, audio sensor to hear when to put down new cards, camera to see the guests and know where to allocate the cards.

1b)

Partially observable – the robot doesn't know the strategies of the guests, and how many cards they are going to desire per turn.

Multi-agent – The people at the table are also trying to maximise their own profits, which means this environment there are multiple people trying to maximise their utility.

Stochastic – Environment changes are based on the actions of the other players rather than on the dealer.

Sequential – The number of the cards that the robot dealer can deal per turn depends on how many are dealt earlier which will affect what suits and what number combinations are possible to obtain.

Static – While the robot is dealing or responding to a situation the players cannot perform any action so the environment remains unchanged.

Discrete – There are a set of rounds with repeated ations for the robot. The robot is confined to the set of moves as the dealer.

2a)

P: The amount of time that the user remains on the system, and if the user engages with the content for long periods of time.

E: The user with who the agent interacts with, and the platform on which the recommendation system operates on.

A: Intelligent algorithms within the software which respond to the actions of the user on the platform. The code processes user action to then create a response.

S: Buttons, search bars, a sensor for time spent that are all part of the platform which allows the system to understand what the user wants to watch. This is all then used to build historical data which is the greatest sensor for the recommendation system.

2b)

Partially observable – The system doesn't know what the user wants to do until it interacts with the software. While it has access to the history, that may not be an accurate depiction of what the user wants to watch based on their current or changed interests. It also doesn't know if the user is not actually engaging with the platform and doing other things while the content is playing.

Multi-agent – You could argue that the user is also a maximising agent because they are trying to maximise the enjoyment from what they view.

Stochastic – The user may not choose to view what the recommender is choosing to do.

Sequential – Recommendation depends on the user's actions to the previous recommendations.

Dynamic – While the machine is generating recommendations the user can decide to be interested in other sorts of content.

Discrete – The type of things that can be recommended are finite, although can be extremely large.

3a)

If we choose n = 2 to be the value of n, the size of the state-space is: S = {{0}, {1}, {1,2), {2}}
This is 4 values.

If we choose n = 3 to be the value of n, the size of the state-space: S = {{0}, {1}, {2}, {3}, {1,2}, {1, 3}, {2, 3}, {1,2,3}}
This is 8 values.

From these two values there is a relationship between the value of n, and the size of the state space. Therefore the largest size of a non-repeating state-space in terms of big-O notation is: 2^n

3b)

The start state is starting with an empty collection.

The goal test has to check if the combination that its currently checking is the largest profit maximiser within the limit. It can do this by checking if the combination matches the one that is determined by the magical oracle.

3c)

In the state space graph, each node appears only once whereas in the tree, nodes and branches can appear multiple times. It is true that multiple nodes can satisfy the goal test, but it does matter which one we choose as you want to use an algorithm that will optimise the time and space taken to find that goal node.

4a)

Expanded: A -> B -> D -> G
This is also the path solution.
If we define the 'optimal' solution to take the shortest path, as cost doesn't matter for DFS, then this is not the optimal path, and the optimal path would be: A -> C -> F

4b)

Expanded: A -> B -> C -> D -> F
This is also the path solution.

This is not the optimal path, as there is a shorter path which is A->C->F

4c)

Expanded path: A->B->C->D->G
Path solution: A->B->D->G

This the optimal path because it is the path that minimises the cost of each edge between the nodes.