Name: Kaimuzzaman (扎尔曼) ID: 104603498612

(1)

1	2	3	4	5
d	b	a	a	c

ID: 1946034812

2.

There are many different kinds of agents:

- Simple reflex agent
- Model based agent
- Goal based agent
- Utility based agent

Above those agent I would choose to design Utility based agent. The reasons I choose it are given below:

i) These agents are flexible and adaptive to the changing environments.

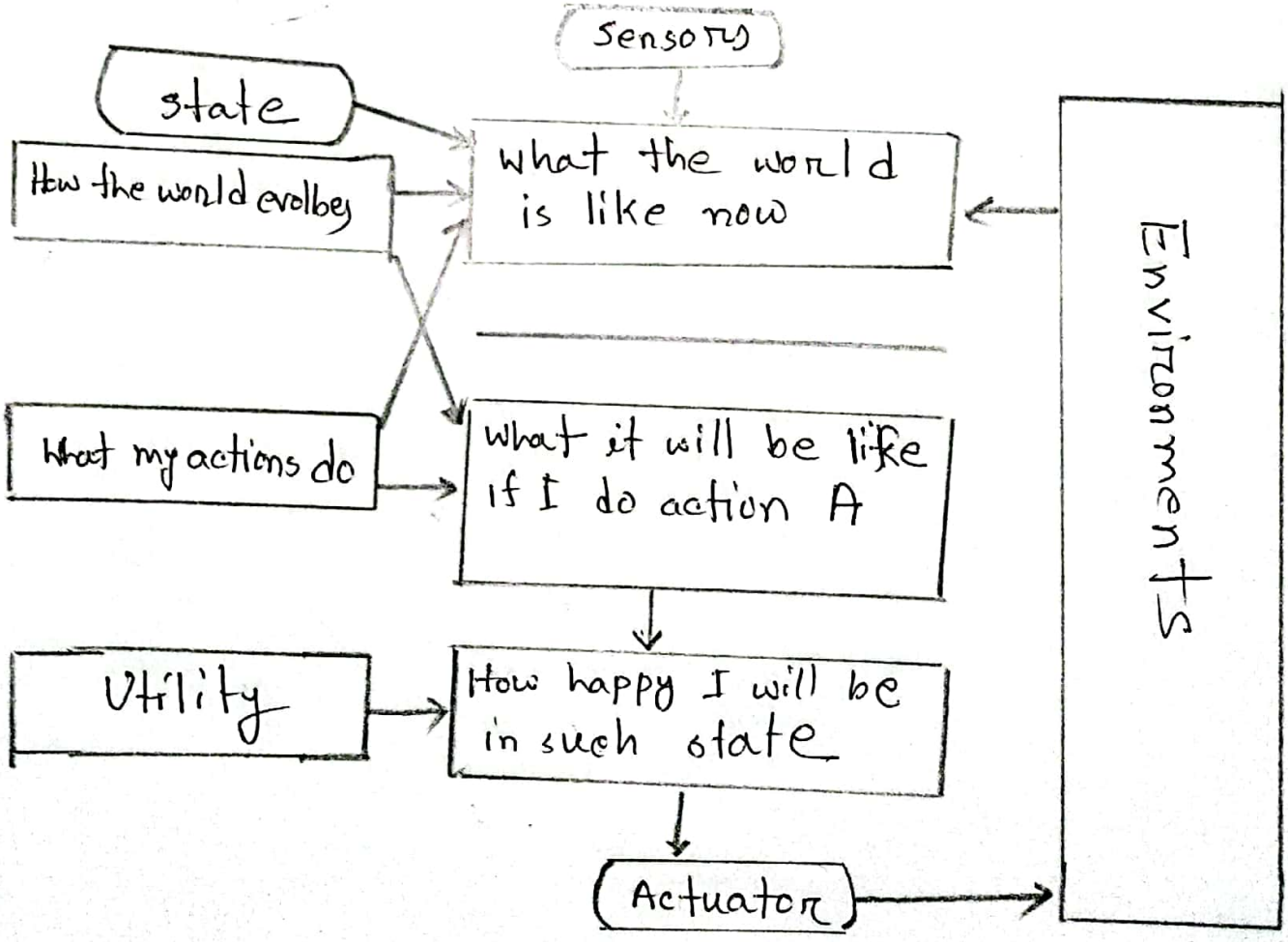
ii) Utility based agents help to choose the best alternatives when there are multiple alternatives available.

iii) A utility function maps a state to measure of the utility of that state.

iv) Based on the external performance measure these agents maintain a high utility function that agent tries to maximise.

v) When there are conflicting goals or several goals that can not be achieved with certainity, these agents can make rational decisions.

The diagram of Utility based agen is given below:



ID: 10460348612

2.2/

To avoid the stuck in local extremum GA uses mutation for individual of generation. Although will case the result worse and create many bad situation for that individual will not be selected for the next generation. But some time mutation get a result which is close to the global minimus which lead to the not stucking in local extremum. The higher the mutation rate the more space will be searched and it will encreas the chance to reach global minimum.

ID: 1046034861 2

2.3

For Wumpus game the list of general steps before moving next step:

I) We already know [1,1] is safe. And here agent doesn't feel any breeze or stench so [1, 2] and [2, 1] is also safe.

II) Agent move [2,1], here it feels breeze ~~stench~~ so, either in [3, 1] or [2, 2] there is a Wumpus. So it go back to [1, 1] → [1, 2]

III) In [1, 2] it feels stench ~~breeze~~ that means in either [1, 3] or [2, 2] there is a pit but from step (II) we know in [2, 2] could have been a ~~wumpus~~ breeze and now a ~~pit~~ wumpus, so we can decide [2, 2] is safe because Wumpus and pit can not be in the same square

ID: 104603486 12

iv) Agent goes to [2,2] it does not get any stench or breeze, so lets uppose it goes [2,3] and perceive glitten so it should grab gold and climb out.

ID: 1046039861 2

2.A|

Among Breath-first, Dept-first, Greedy search, and A* search, I will choose A* search to solve 8-puzzle. Because A* uses a combination of heuristic value (h(n): how far the goal node is) as well as g(n) (i.e the number of nodes traversed from the start node to current node). The key feature of the A* algorithm is that it keeps a track of each visited node which help to ignore the nodes that already visited, saving a huge amount of time. It also has a list that holds all the nodes that are left to be explored and it choose the most optimal node from this node list, thus saving time not exploring unnecessary or less optimal nodes.

The difference between A* and Greed Search:

A* search	Greed search
1) A* is complete and optimal	It is often not optimal but often efficient
11) It expand node with minimal $f(n) = g(n) + h(n)$	It expand node with only $f(n) = g(n)$

ID: 1046034612

The difference between A* and BFS/DFS :

A* search	BFS/DFS
i) It uses knowledge for searching process	Does not use knowledge for searching process
ii) It finds solution more quickly	It finds solution slow as compared to informed search
iii) Cost is low	cost is high
iv) It consumes less time	It consume more time than A*
v) It is less lengthy while implementation	While implementation it is more lengthy

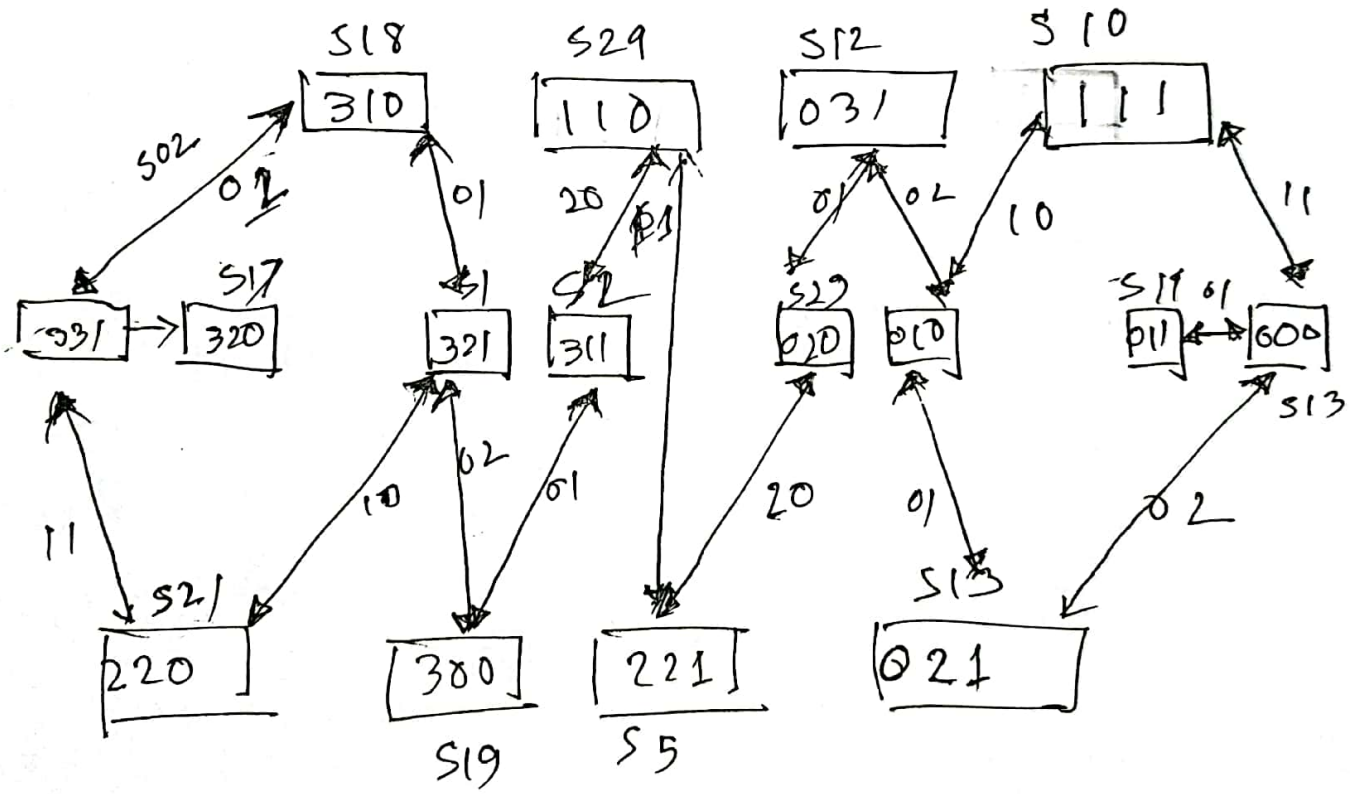
ID: 1096034861

2.5

The initial state is: [3, 3 1]

Final state is: [0, 0, 0]

The state space of missionar and cannibal



ID: 1046034861 2

3.1 |

(a) UCS :

Expnd. node	Open list
	{S:0}
S	{A:1}
A	{B:1+1=2, D:3+1=4, E:8+1=9}
B	{D:4, E:9, C:1+1+1=3}
C	{D:4, E:9}
D	{E:9, G:1+3+2=6}
G (goal)	{E:9} (not expand)

Path: S, A, D, G

Cost: 6

b) Greed search

Expnd. node	Open list
	{S:0}
S	{A:1}
A	{B:1, D:3, E:8}
B	{C:1}
C (fail to find the goal)	

ID: 1046639861 2

c) A* search:

Expnd. node	Open list
	{ S: 0+6 }
S	{ A: 1+5=6, B:
A	{ B: 1+1+6=8, D: 1+3+2=6, E: 1+8+1=10 }
D	{ B: 8, E: 10, G: 1+3+2+0 = 6 }
G (goal)	{ B: 8, E: 10 } (not Expand)

Path: S, A, D, G

Cost: 6

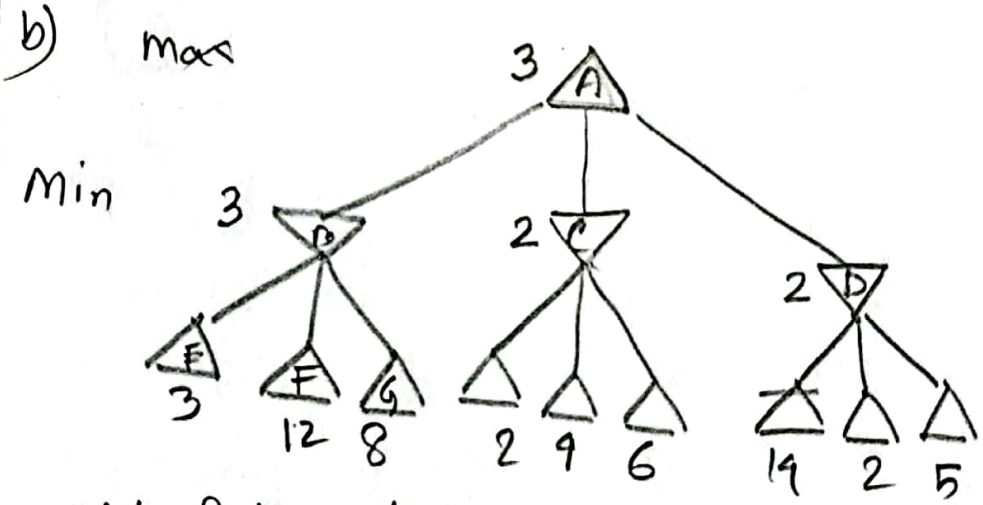
D) The least cost path from S to G is: S → A → D → G

ID: 1048034812

3.2

(a) In two players game the first player is called max because he always try to choose the value which will maximize his win on the other hand the second player is called min because he always tray to choose the value which will minimize the first players win probability so that he can win.

b)



List of the values

Node:	B	C	D	A
Value	3	2	2	3

The final value is 3

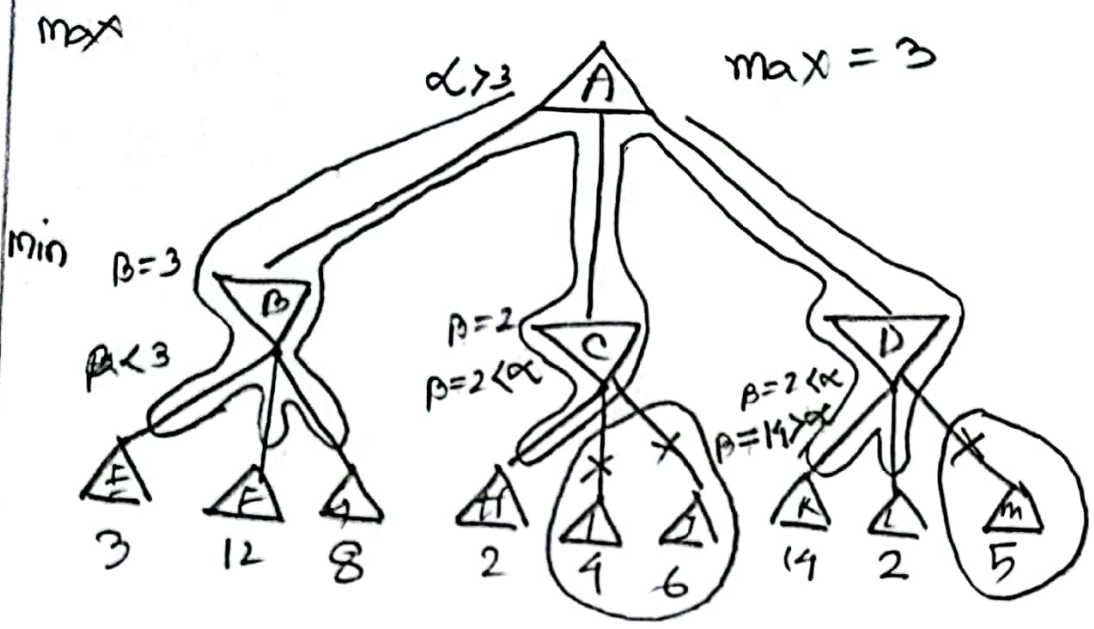
c) The final value of the game can not be 14 or 2 reasons are below:

For 14, we can see the value 14 is can ~~found~~ be found A → D → K but as the D is min so it will choose the minimum value among its other node. Othe nodes L = 2, m = 5 so D will never choos 14 as its value.

On the other hand, for 2, it can be found A → C → H and A → D → L, so if we see the other node, 2 is the minimum value in both C and D so it will choose it but when it comes to A it will choose the maximum value among B, C, D. In B, it can find 3 which is greater than 2, so 2 can not be the final value.

ID: 1096634861

d)



I, J and M node would not be visited if alph-beta pruning is used

ID: 1046034612

3.3

a) The equation for y_1, y_2 and z is given below:

$$y_1 = 0.5x_1 - x_2 + 1$$

$$y_2 = -x_1 - x_2 + 2$$

$$z = y_1 - y_2 - 0.5$$

x_1	x_2	y_1	$y_1>0$	y_2	$y_2>0$	z	$z>0$
0	0	1	1	2	1	-0.5	0
0	1	0	0	1	1	-1.5	0
1	0	1.5	1	1	1	-0.5	0
1	1	0.5	1	0	0	0.5	1

b) The boolean function it implement is $y_1 \cdot \overline{y_2}$.

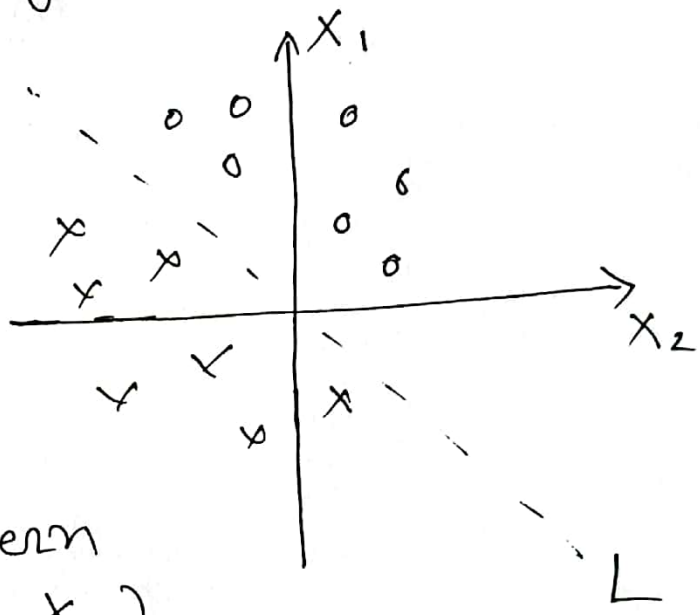
y_1	y_2	$\overline{y_2}$	$z = y_1 \cdot \overline{y_2}$
1	1	0	0
0	1	0	0
1	1	0	0
1	0	1	1

ID: 1096034862

3.3

The results of XOR operator can not be separated by single layer because it is not linearly separable.

Conside two-input patterns (x_1, x_2) being classified into two classes as shown in the right



x, o repnents a pattern with a set value of (x_1, x_2)

Notice that L a line separate the classes. They are known as linearly separable patterns. Two classes must be separable in order for the perception network function correctly.

- 1 layer perception ≡ 1 dimensional separator

Therefore, since XOR output can not be separated wing a staigh line, they can not be classified wing single-layer perceptron.