Q1] a) True.

b) False

c) False

Because regression is used to predict continous value

d) True

e) False.

K-means clusters, the data points with different centroids. Concentric circle have same radius for different categories of data. which cannot be classified / categorised using K-means algorithm

f) False.

An underfit model cannot learn the underlying patterns in the data. which destroys the accuracy of the model.

g) True

h) True

learning rate must traditionally be

choosen between 0.1 to 0.001 and then should be tuned via trial and error.

i) True

j) True

k) False

Backpropagation works by propagating gradient loss from OUTPUT layer to INPUT layer

L) True (T)

m) $(-1 \times w) + (-1 \times w) + (-1 \times w) + (-1 \times w) = -1$

$-w - w - w - w = -1$

$-4w = -1$

if $w = 1$

$Relu(-4 \times 1)$

$Relu(-4)$ $\{0 - \infty\}$

False

n) True.

o) False

gradient $\begin{cases} 0 & \text{if } x < 0 \\ 1 & \text{if } x > 0 \end{cases}$

a)

Q2) Criteria.

i) $h(\text{successor}(s)) - h(s) \leq \text{cost}(s, \text{succesor}(s))$

ii) $h(\text{goal}) = 0$

Q2) b) i) The algorithm is used to choose an optimal move for a player assuming that the other player is also playing optimally.

ii) The algorithms should help in minimizing the loss when the other player chooses the strategy having the maximum loss

c) A search algorithm is complete if it is able to find atleast one solution. It is guaranteed that the search algorithm will

find a solution.

If a search algorithm is optimal then, it means when the algorithm finds a solution it finds the best solution.

2)d) The memory usage for

BFS : $O(b^d + 1)$

IDS : $O(bd)$

where d is depth.

IDS does not store all the visited nodes. it only stores the path and the nodes that it has expanded on the path. and the other nodes are disgarded hence consuming less memory.

In BFS, we dont know where the solution is, so we cannot disgard the nodes

2e) In alpha beta pruning, we prune half of the search space, which is not desirable for solving maze problem.
Algorithms like BFS is well suited for such problems which considers the entire search space.

2f) The DFS algorithms has to reach reach till the leaf node in order to terminate whereas as backtracking stops if the constraint is not met.
DFS is used for tree structure whereas as backtracking is a genral purpose algorithm.

In Backtracking the nodes are only expanded if they are promising, in the sense of being feasible precursors to a solution. which is why one might prefer Backtracking over DFS

g) Eg: When in search, the agents are designed to move at any angle instead of in a grid.
then Euclidean distance is not a good heuristic

h) To make h(B) admissible
h(B) ≤ optimal cost (B → C) which is
if cost (A → C) : 0
Ans: 0 = h(B) = 0

2 i) def isEnd()

if pos_x == goal_x and

pos_y == goal_y

return true.

return False

2 j) def isSuccessor(i, j):

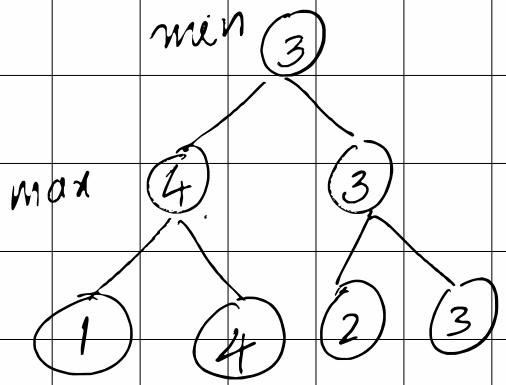
if pos_x + 1 == i and

pos_y + 1 == j

return true.

return false

k)



$A = 3$

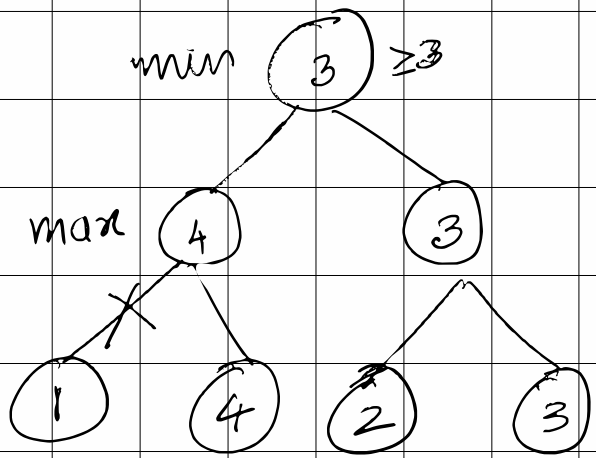
$B = 4$

$C = 3$

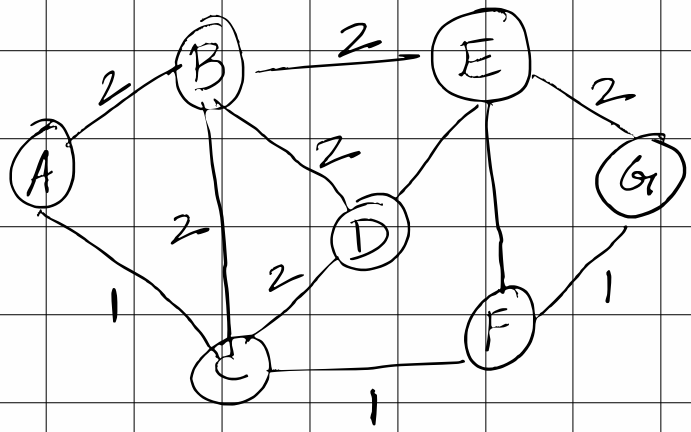
l) In this example we cannot prune any edges.

right child of c > left child of c
& c is a maximizer function

m) In this case we can prune



o)



	Path	Algo
n)	B-E-G	DFS
	A-B-C	BFS

Q3) Entropy (Bus)

$$= \sum P_i \log(P_i)$$

$$= -\frac{7}{10} \log \frac{7}{10} - \frac{3}{10} \log \frac{3}{10}$$

$$= -0.7 \log 0.7 - 0.3 \log 0.3$$

$$= 0.10843 + 0.15686$$

$$= 0.2653$$

b) Entropy (bus, rain)

$= \mathcal{E}(\text{bus, low}) +$

$\mathcal{E}(\text{bus, high}) +$

$\mathcal{E}(\text{bus, medium})$

$= -\left[\frac{2}{3}\log\frac{2}{3} + \frac{1}{3}\log\frac{1}{3}\right]$

$- \left[0\right] - \left[\frac{1}{3}\log\frac{1}{3} + \frac{2}{3}\log\frac{2}{3}\right]$

$0.6365 + 0 + 0.6365$

1.2730

3.c) $E(\text{Bus, far})$

$$= -\left[\frac{5}{10}\log\frac{5}{10} + \frac{5}{10}\cdot\log\frac{5}{10}\right]$$

$$= 0.69314$$

3 d) First ~~feature~~ to be selected is "Far"

3 e) 2nd feature: rain.

4) a) Cluster 1 = (1.90, 0.97)

Cluster 2 = (5.74, 3.60)

as they clearly form 2 clusters.

b) using euclidian distan

C1	0	0.19	0.78	3.73	4.65	4.07
C2	4.65	4.84	3.94	0.92	0	0.97

assigning to the nearest cluster.

	A	B	C	D	E	F	
C1	1	1	1	0	0	0	G1 ⇒ ABC (1 2 3)
C2	0	0	0	1	1	1	G2 ⇒ DEF (4, 5, 6)

c) centroid is the average distance among the cluster members

$$c_1 = \frac{1.90 + 1.76 + 2.32}{3}, \frac{0.97 + 0.84 + 1.63}{3}$$

$$c_1 = (1.99, 1.46)$$

$$c_2 = \frac{5.02 + 5.74 + 4.79}{3}, \frac{3.02 + 3.60 + 3.84}{3}$$

$$c_2 = (5.183, 3.48)$$

d) new point = 5.20, 7.51

distance from centroid c_1 = 7.12

" " " c_2 = 4.030

e) Here we can use cross fold validation to determine noise

f) ii) p_2 (5.0, 4.0) is correct reading