Assignment 1 - COMP 6721

Applied Artificial Intelligence

Group Name: AI_Bots
Group Members:

1. Huzaifa Mohammed (40242080),
2. Mohammed Shurrab (40323793),
3. Oleksandr Yasinovskyy (40241188)

All members have contributed equally to the solution of the Assignment

Professor
Prof. Arash Azafar

Date
01/06/2025

Question 1:

Travelling back to Expo 67, a time of massive technological optimism, armed with today's knowledge from the Applied AI course, I would be in a unique position to influence the early trajectory of AI development and possibly help prevent the first AI winter of the 1970s.

a) Whom would you visit?

List of Key AI Experts around 1967:

- Marvin Minsky – Co-founder of the MIT Artificial Intelligence Laboratory and a key proponent of symbolic AI.
- John McCarthy – Coined the term "Artificial Intelligence" and developed LISP, the programming language foundational to early AI.
- Allen Newell – Pioneer in cognitive simulation and symbolic AI; worked with Simon on the Logic Theorist and General Problem Solver (GPS).
- Herbert A. Simon – Nobel Prize winner and co-developer of some of the first AI programs.
- Claude Shannon – Father of information theory; though not solely focused on AI, his work underpins modern machine learning.
- Norbert Wiener – Founder of cybernetics, which heavily influenced early AI thinking.
- Joseph Weizenbaum – Creator of ELIZA; later became critical of overhyping AI capabilities.

I would visit Marvin Minsky a leading AI researcher at the time but also someone with strong influence in academia and funding circles. Talking to him could have had a ripple effect on how the field evolved.

b) What would you suggest to Minsky? How could you help prevent the AI Winter?

1. Warn Against Overpromising and Underdelivering

"Marvin, one of the biggest threats to the future of AI is not failure—it's overconfidence."

I will explain that bold claims in the late '60s (like human-level AI in 10 years) will lead to inevitable disillusionment when results don't match expectations.

I will suggest being honest about timelines and focusing on small, steady progress.

2. Introduce the Concept of Machine Learning (ML) and Neural Networks

While symbolic AI was dominant, neural networks were still in their infancy and largely dismissed due to limitations in computing power and theory (e.g., the XOR problem).

I will suggest sharing the future development of backpropagation and how it solves the XOR problem. I will also give early insights into deep learning, gradient descent, and CNNs to guide future research.

3. Encourage Data-Driven Approaches

I will explain how the success of future AI models (like ChatGPT!) will depend heavily on large datasets, computational power, and statistical learning rather than only symbolic logic.

I will suggest introducing the importance of probabilistic models, learning from data, and working across fields like statistics and neuroscience.

4. Help Build Sustainable AI Research Practices

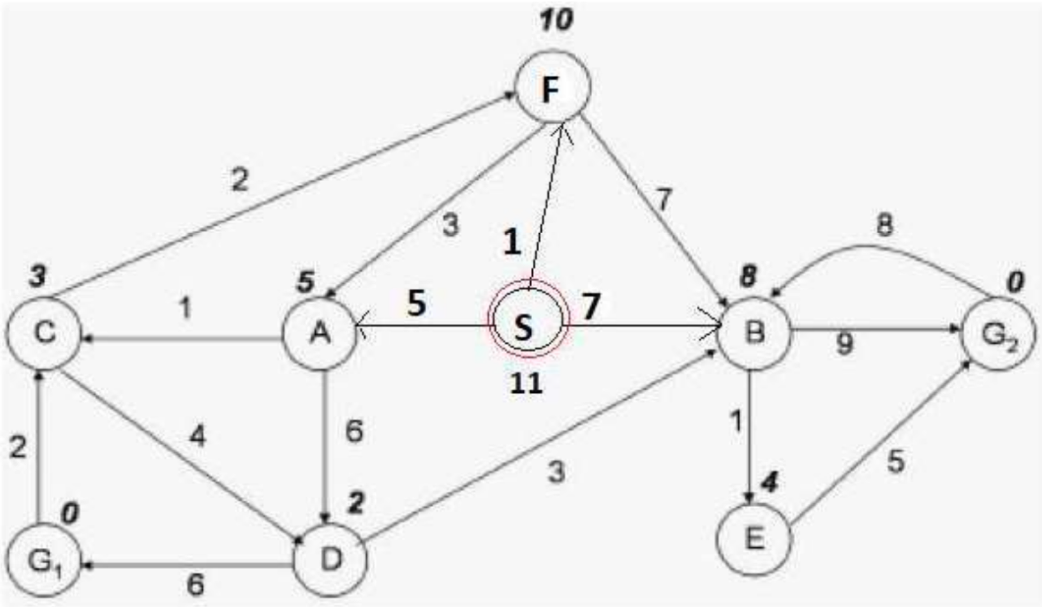
I will suggest providing a roadmap to avoid cycles of hype and disappointment by supporting diverse funding, promoting open-source collaboration, and focusing on realistic goals like domain-specific AI.

5. Provide a Glimpse of the Future

“In 2025, models with billions of parameters will generate human-like text, compose music, and even code.”

I will share future use-cases like GPT, self-driving cars, AlphaGo, and recommendation engines to guide research focus, highlight the value of neural and statistical AI, and keep policymakers and funders engaged.

Question 2:



a. Breadth-first search

Step	Visited	Closed List	Open List
1			S
2	S	S	A, B, F
3	A	S, A	B, F, C, D
4	B	S, A, B	F, C, D, E, G2
5	F	S, A, B, F	C, D, E, G2
6	C	S, A, B, F, C	D, E, G2
7	D	S, A, B, F, C, D	E, G2, G1
8	E	S, A, B, F, C, D, E	G2, G1
9	G2	S, A, B, F, C, D, E, G2	G1

The path to the goal is: S > B > G2; Cost = 16.

b. Depth-first search

Step	Visited	Closed List	Open List
1			S
2	S	S	A, B, F
3	A	S, A	C, D, B, F
4	C	S, A, C	D, B, F
5	D	S, A, C, D	G1, B, F
6	G1	S, A, C, D, G1	B, F

The path to the goal is: S > A > D > G1; Cost = 17.

c. Iterative deepening depth-first search

Depth 1:

Step	Visited	Closed List	Open List
1			S (1)
2	S (1)	S (1)	

Depth 2:

Step	Visited	Closed List	Open List
1			S (1)
2	S (1)	S (1)	A (2), B (2), F (2)
3	A (2)	S (1), A (2)	B (2), F (2)
4	B (2)	S (1), A (2), B (2)	F (2)
5	F (2)	S (1), A (2), B (2), F (2)	

Depth 3:

Step	Visited	Closed List	Open List
1			S (1)
2	S (1)	S (1)	A (2), B (2), F (2)
3	A (2)	S (1), A (2)	C (3), D (3), B (2), F (2)
4	C (3)	S (1), A (2), C (3)	D (3), B (2), F (2)
5	D (3)	S (1), A (2), C (3), D (3)	B (2), F (2)
6	B (2)	S (1), A (2), C (3), D (3), B (2)	E (3), G2 (3), F (2)
7	E (3)	S (1), A (2), C (3), D (3), B (2), E (3)	F (2)
8	G2 (3)	S (1), A (2), C (3), D (3), B (2), E (3), G2 (3)	

The path to the goal is: S > B > G2; Cost = 16.

d. Uniform cost search (UCS)

Step	Visited	Closed List	Open List
1			S (0)
2	S (0)	S (0)	F (1), A (5), B (7)
3	F (1)	S (0), F (1)	A (4), B (7)
4	A (4)	S (0), F (1), A (5)	C (5), B (7), D (10)
5	C (5)	S (0), F (1), A (5), C (5)	B (7), D (9)
6	B (7)	S (0), F (1), A (5), C (5), B (7)	E (8), D (9), G2 (16)
7	E (8)	S (0), F (1), A (5), C (5), B (7), E (8)	D (9), G2 (16)
8	D (9)	S (0), F (1), A (5), C (5), B (7), E (8), D (9)	G2 (13), G1 (15)
9	G2 (13)	S (0), F (1), A (5), C (5), B (7), E (8), D (9), G2 (13)	G1 (15)

The path to the goal is: S > B > E > G2; Cost = 13.

e. Hill climbing

Step	Visited	Evaluations
1	S	A (5) < S (11)
2	A	C (3) < A (5)
3	C	D (2) < C (3)
4	D	~~B (8) > D (2)~~; G1 (0) < D (2)
5	G1	

The path to the goal is: S > A > C > D > G1; Cost = 16.

f. Best-first search

Step	Visited	Closed List	Open List
1			S (11)
2	S (11)	S (11)	A (5), B (8), F (10)
3	A (5)	S (11), A (5)	D (2), C (3), B (8), F (10)
4	D (2)	S (11), A (5), D (2)	G1 (0), C (3), B (8), F (10)
5	G1 (0)	S (11), A (5), D (2), G1 (0)	C (3), B (8), F (10)

The path to the goal is: S > A > D > G1; Cost = 17.

g. Beam search with ß=2

Step	Visited	Closed List	Open List
1			S (11)
2	S (11)	S (11)	A (5), B (8)
3	A (5)	S (11), A (5)	D (2), C (3)
4	D (2)	S (11), A (5), D (2)	G1 (0), C (3)
5	G1 (0)	S (11), A (5), D (2), G1 (0)	C (3)

The path to the goal is: S > A > D > G1; Cost = 17.

h. Algorithm A

Step	Visited	Closed List	Open List
1			S (11)
2	S (11)	S (11)	A (10), F (11), B (15)
3	A (10)	S (11), A (10)	C (9), F (11), D (13), B (15)
4	C (9)	S (11), A (10), C (9)	F (11), D (12), B (15)
5	F (11)	S (11), A (10), C (9), F (11)	D (12), B (15)
6	D (12)	S (11), A (10), C (9), F (11), D (12)	B (15), G1(16)
7	B (15)	S (11), A (10), C (9), F (11), D (12), B (15)	E (12), G1 (16), G2 (16)
8	E (12)	S (11), A (10), C (9), F (11), D (12), B (15), E (12)	G2 (13), G1 (16)
9	G2 (13)	S (11), A (10), C (9), F (11), D (12), B (15), E (12), G2 (13)	G1 (16)

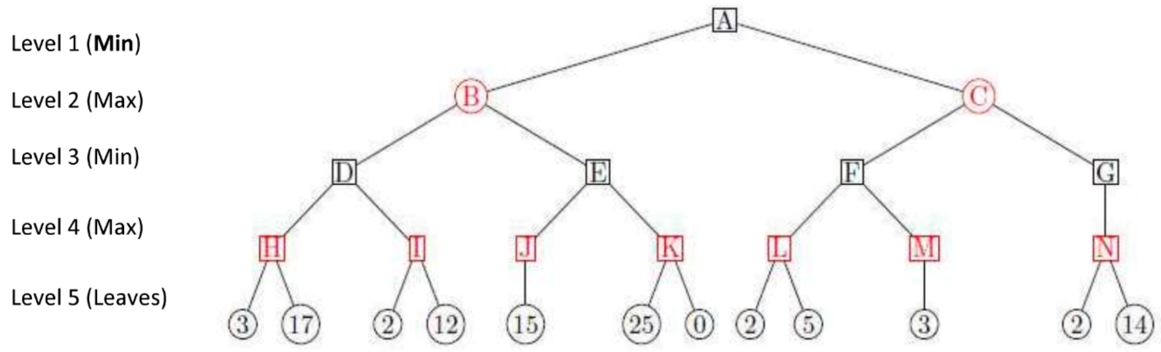
The path to the goal is: S > B > E > G2; Cost = 13.

i. With this heuristic, are A and A* equivalent?

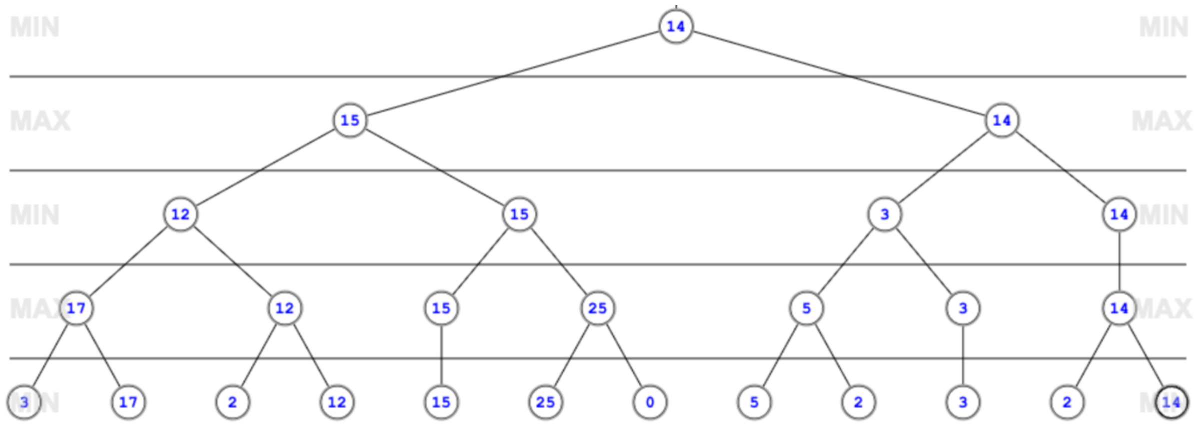
Algorithm A is a best-first search algorithm that selects nodes for expansion based on the evaluation function F(n)=g(n)+h(n), where g(n) is the cost from the start node to the current node and h(n) is a heuristic estimate of the cost from that node to the goal. While Algorithm A places no strict requirements on the heuristic function h(n), it assumes that $g(n) \geq g^*(n)$, meaning the cost incurred to reach a node is at least the optimal cost. Algorithm A* is a specialized version of Algorithm A in which the heuristic function is admissible; that is, $h(n) \leq h^*(n)$ for all nodes n, where $h^*(n)$ represents the true minimal cost from node n to the goal. An admissible heuristic never overestimates the cost to reach the goal, ensuring that A* always finds an optimal solution. In the provided graph, all heuristic values are non-negative, the heuristic values at the goal nodes G1 and G2 are zero, and the heuristic does not overestimate the actual cost to reach the goal. This confirms that the heuristic is admissible. As a result, in this specific case, Algorithm A and A* are equivalent in behaviour and will both find the optimal (lowest-cost) solution path.

Hence, the path S > B > E > G2 is the optimal path to the goal with the cost of 13.

Question 3:

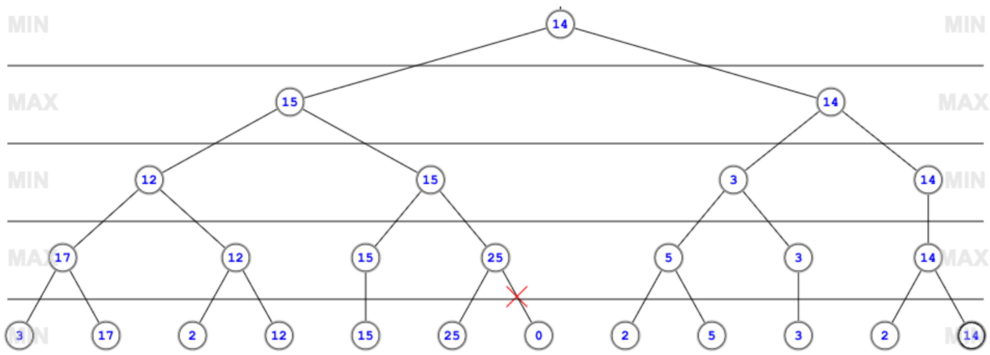


a. [3 pts] Run minimax. What will be the value of the root (A)? Hint: The first player is a minimizer. Intermediate values should be provided.



The value of root A is 14.

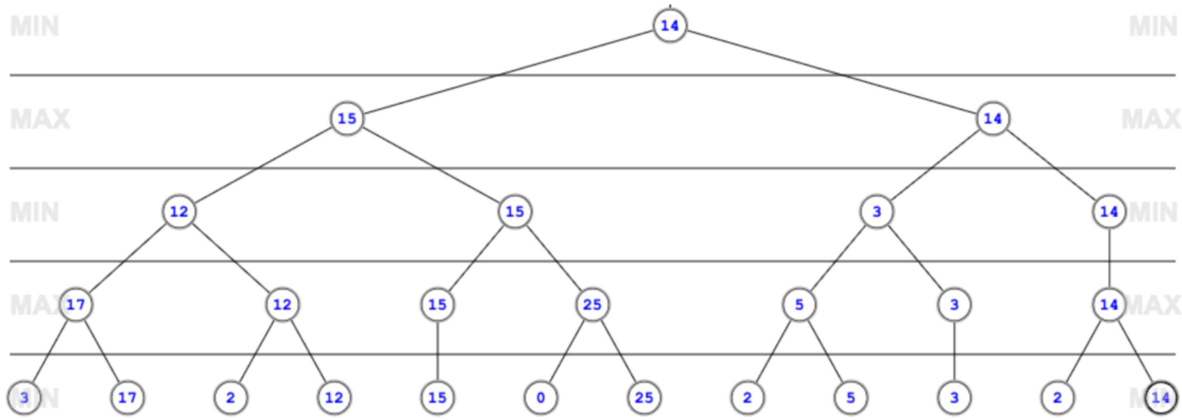
b. [4 pts] After running minimax, indicate the branches that will be pruned with alpha-beta pruning?



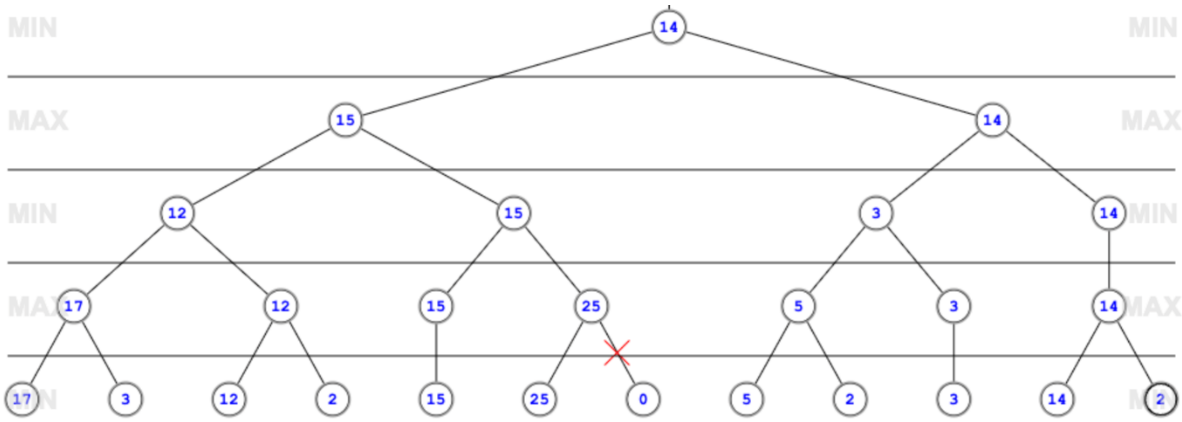
The edge K-0 will be pruned.

c. [3 pts] Now assume all the games with this structure and any possible values on the leaves. In the best and worst case, how many leaves will be pruned by alpha-beta pruning?

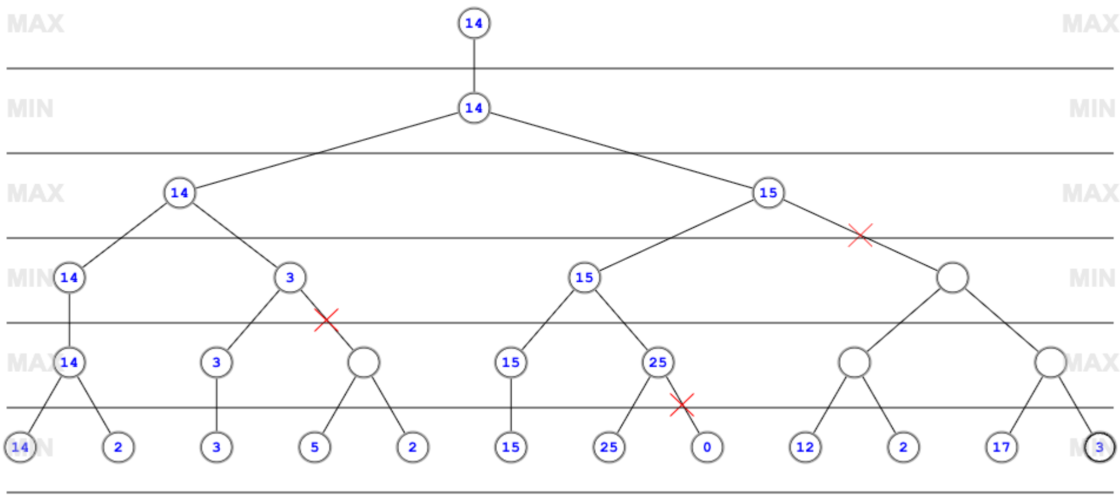
Worst case: No nodes are pruned



Best case: 1 node is pruned. Assuming that we can only switch the leaf nodes and not change the structure.



Best case (with changing structure): 11 nodes are pruned. Assuming that we can switch the leaf nodes and change the structure.



Question 4:

a)

Q4) Probability when :

C : cinema
T : Tennis
SI : Stay-in

weather = sunny , Cash = poor , Exam = no

* Priors : $P(C) = \frac{6}{12} = 0.5$, $P(T) = 3/12 = 0.25$, $P(SI) = 3/12 = 0.25$

* Posteriors :

$P(\text{sunny} \mid C) = 1/6$, $P(\text{sunny} \mid T) = 3/3$, $P(\text{sunny} \mid SI) = 0/3 = 0$

∴ Stay in will be zero.

$P(\text{poor} \mid C) = 3/6 = 0.5$, $P(\text{poor} \mid T) = 1/3$, $P(\text{poor} \mid SI) = 0/3 = 0$

$P(\text{No} \mid C) = 3/6 = 0.5$, $P(\text{No} \mid T) = 2/3$, $P(\text{No} \mid SI) = 2/3$

- Cinema : $P(C) \times P(\text{sunny} \mid C) \times P(\text{poor} \mid C) \times P(\text{No} \mid C)$

$= 0.5 \times 0.167 \times 0.5 \times 0.5$

$= 0.0208$

- Tennis : $P(T) \times P(\text{sunny} \mid T) \times P(\text{poor} \mid T) \times P(\text{No} \mid T)$

$= 0.25 \times 1 \times 0.333 \times 0.667$

$= 0.056$

- Stay-In : since there is a probability that is equal to zero

$= 0$

Normalize :

$\text{Cinema} = \frac{0.0208}{0.0208 + 0.056} = \boxed{0.27}$

$\text{Tennis} = \frac{0.056}{0.0208 + 0.056} = \boxed{0.73}$

Stay-in = 0

Hence , Tennis is the most probable outcome with probability of 0.73 (73 %)

b)

b) if priors are $P(C) = 0.4$, $P(T) = 0.4$, $P(SI) = 0.2$

The likelihood will not change but the final probabilities will change

Cinema : $0.4 \times 0.167 \times 0.5 \times 0.5 = 0.0167$

Tennis : $0.4 \times 1 \times 0.333 \times 0.667 = 0.0888$

→ Normalize :

$$\text{Cinema} = \frac{0.0167}{0.0167 + 0.0888} = \boxed{0.158}$$

$$\text{Tennis} = \frac{0.0888}{0.0167 + 0.0888} = \boxed{0.842}$$

The final outcome is still the same but we are more certain now with a probability of 0.842 (84.2%)

c)

```
Names (ID): Huzaifa Mohammed (40242080), Mohammed Shurrab (40323793),
 Oleksandr Yasinovskyy (40241188)
With Smoothing
Classes =  ['Cinema' 'Stay-in' 'Tennis']
Predicted class =  ['Tennis']
Predicted class probability=  [[0.24753475 0.04426266 0.70820259]]
```

```
No Smoothing
Classes =  ['Cinema' 'Stay-in' 'Tennis']
Predicted class =  ['Tennis']
Predicted class =  [[2.72727273e-01 2.42424242e-21 7.27272727e-01]]
```

Question 5:

1) a)

Q5 1)

a) $H(S) = -\sum P_i \log_2 P_i = -[P(C)\log_2(P(C)) + P(T)\log_2(P(T)) + P(SI)\log_2(P(SI))]$

$= -\left[\frac{6}{12}\log_2(6/12) + \frac{3}{12}\log(3/12) + \frac{3}{12}\log(3/12)\right]$

$= 1.5$

for weather :

sunny : C=1 , T=3, SI=0

windy : C=3 , T=0, SI=1

rainy : C= 2 , T=0, SI=2

$H(S|\text{sunny}) = H(1/4, 3/4, 0) = -\left[\frac{1}{4}\log_2(1/4) + \frac{3}{4}\log_2(3/4) + \cancel{\frac{0}{4}\log_2(0/4)}\right]$

$= 0.811$

$H(S|\text{windy}) = H(3/4, 0, 1/4) = -\left[\frac{3}{4}\log_2(3/4) + 0 + \frac{1}{4}\log(1/4)\right]$

$= 0.811$

$H(S|\text{rainy}) = H(2/4, 0, 2/4) = -\left[\frac{2}{4}\log(2/4) + 0 + \frac{2}{4}\log(2/4)\right]$

$= 1$

$H(S|\text{weather}) = \frac{4}{12}H(S|\text{sunny}) + \frac{4}{12}H(S|\text{windy}) + \frac{4}{12}H(S|\text{rainy})$

$= 0.874$

$\text{gain}(\text{weather}) = H(S) - H(S|\text{weather}) = 1.5 - 0.874 = \boxed{0.626}$

Attribute = Cash

rich : C = 3 , T= 2, SI= 3

poor : C= 3 , T= 1 , SI=0

$H(S|\text{rich}) = H(3/8, 2/8, 3/8) = -\left[\frac{3}{8}\log_2(3/8) + \frac{2}{8}\log_2(2/8) + \frac{3}{8}\log_2(3/8)\right]$

$= 1.5$

$H(S|\text{poor}) = H(3/4, 1/4, 0) = -\left[\frac{3}{4}\log_2(3/4) + \frac{1}{4}\log_2(1/4)\right]$

$= 0.811$

$H(S|\text{cash}) = 8/12\,(1.5) + 4/12\,(0.811) = 1.27$

$\text{gain}(\text{cash}) = 1.5 - 1.27 = \boxed{0.23}$

⊛ Attribute = Exam

Yes : C = 3 , T = 1 , SI = 1

No : C = 3 , T = 2 , SI = 2

$$H(S|yes) = H(3/5, 1/5, 1/5) = -\left[3/5 \log_2(3/5) + 1/5 \log(1/5) + 1/5 \log(1/5)\right]$$

$$= 1.522$$

$$H(S|No) = H(3/7, 2/7, 2/7) = -\left[3/7 \log_2(3/7) + 2/7 \log_2(2/7) + 2/7 \log(2/7)\right]$$

$$= 1.556$$

$$H(S|Exam) = 5/12\,(1.522) + 7/12\,(1.556) = 1.544$$

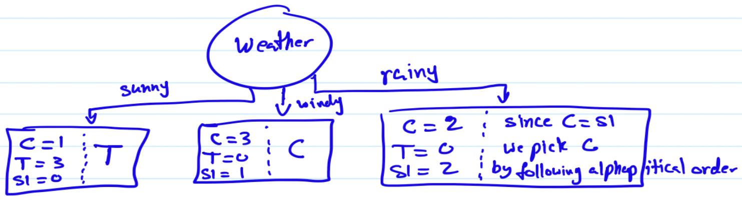
$$gain(Exam) = 1.5 - 1.544 = \boxed{-0.044}$$

since gain(weather) > gain(cash) > gain(Exam)

The best attribute is weather because it has the highest gain (entropy)

b)

b)



given the new instance is (sunny, poor, No) the prediction is Tennis based on our DT

* Performance metrics :

Obs.	1	2	3	4	5	6	7	8	9	10	11	12
Weather	S	S	W	R	R	R	W	W	W	S	S	R
Actual	C	T	C	C	SI	C	C	SI	C	T	T	SI
Predicted	T	T	C	C	C	C	C	C	C	T	T	C
	✗	✓	✓	✓	✗	✓	✓	✗	✓	✓	✓	✗

8 out of 12 are correct, hence:

$$Accuracy = 8/12 = 0.667 = 66.7\%$$

Confusion matrix :

Predicted \ Actual	C	T	SI
C	5	0	3
T	1	3	0
SI	0	0	0

⊛ Cinema

$TP = 5$, $FP = 3$, $FN = 1$,

$$\text{Recall} = \frac{TP}{TP+FN} = \frac{5}{6} = 0.833$$

$$\text{Percision} = \frac{TP}{TP+FP} = \frac{5}{8} = 0.625$$

$$\text{F1.score} = 2\frac{PR}{P+R} = 2.\frac{(0.625)(0.833)}{0.625+0.833} = 0.714 \quad (\text{Assuming } B=1)$$

⊛ Tennis:

$TP = 3$, $FP = 1$, $FN = 0$

$$\text{Recall} = \frac{3}{3} = 1$$

$$\text{Percision} = \frac{3}{4} = 0.75$$

$$\text{F1-score} = 2\frac{0.75}{1+0.75} = 0.857$$

⊛ Stay-in:

$TP = 0$, $FP = 0$, $FN = 3$

$$\text{Recall} = \frac{0}{0} \Rightarrow 0$$

$$\text{Percision} = 0$$

$$\text{F1-score} = 0$$

2)

2) branching factor of 3:

gain(weather) = 0.626 (from part 1)

- branching factor of 2:

There are 3 possibilities
① (sunny, windy & rainy) - case 1
② (windy, sunny & rainy) - case 2
③ (rainy, sunny, windy) - case 3

- case 1:
sunny : C = 1, T = 3, SI = 0
windy + rainy: C = 5, T = 0, SI = 3

H(S | sunny) = 0.811 (from part 1)

$$H(S \mid \text{windy+rainy}) = H(5/8, 0, 3/8) = -\left[5/8 \log_2(5/8) + 3/8 \log_2(3/8)\right]$$
$$= 0.954$$

$$H(S \mid \text{case 1}) = 4/12\,(0.811) + 8/12\,(0.954) = 0.906$$

$$\text{gain(case 1)} = 1.5 - 0.906 = \boxed{0.595}$$

- case 2:
windy : C = 3, T = 0, SI = 1
sunny + rainy: C = 3, T = 3, SI = 2

$$H(S \mid \text{windy}) = H(3/4, 0, 1/4) = 0.811 \text{ (from part a)}$$

$$H(S \mid \text{sunny+rainy}) = H(3/8, 3/8, 2/8) = -\left[3/8 \log_2(3/8) + 3/8 \log_2(3/8) + 2/8 \log_2(2/8)\right]$$
$$= 1.562$$

$$H(S \mid \text{case 2}) = 4/12\,(0.811) + 8/12\,(1.562) = 1.311$$

$$\text{gain (case 2)} = 1.5 - 1.311 = \boxed{0.189}$$

- case 3:

rainy: $C = 2$, $T=0$, $SI = 2$

sunny + windy: $C=4$, $T=3$, $SI = 1$

$H(S|\text{rainy}) = H(2/4, 0, 2/4) = 1$ (from part 1)

$H(S|\text{sunny+windy}) = H(4/8, 3/8, 1/8) = -\left[4/8 \log_2(4/8) + 3/8 \log_2(3/8) + 1/8 \log_2(1/8)\right]$

$= 1.406$

$H(S|\text{case }3) = 4/12\,(1) + 8/12\,(1.406) = 1.271$

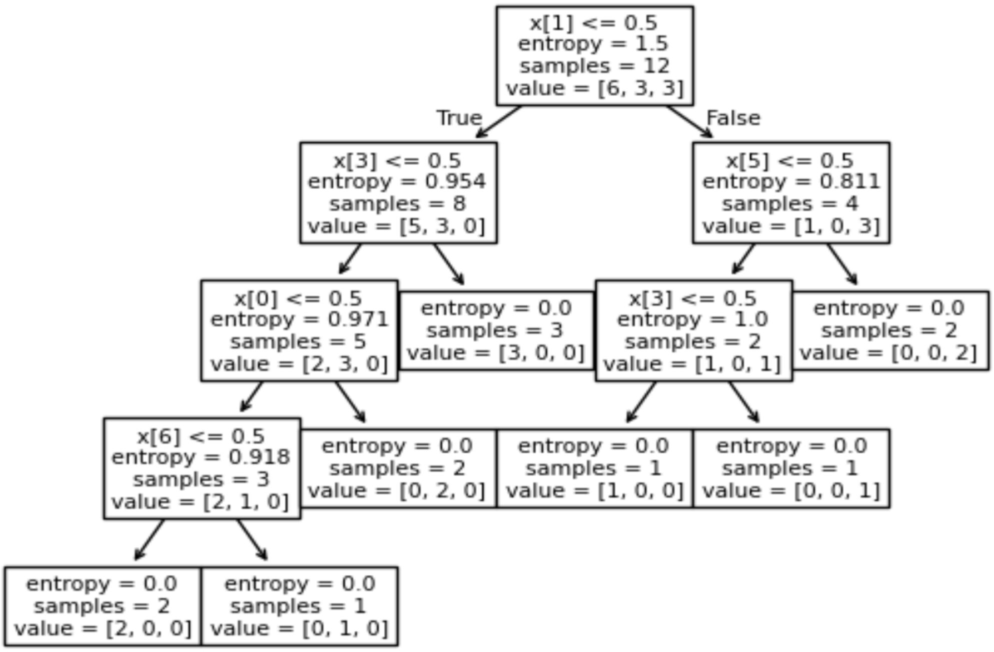
$\text{gain}(\text{case}3) = 1.5 - 1.271 = \boxed{0.229}$

⇒ branching factor of 3 gives higher information gain

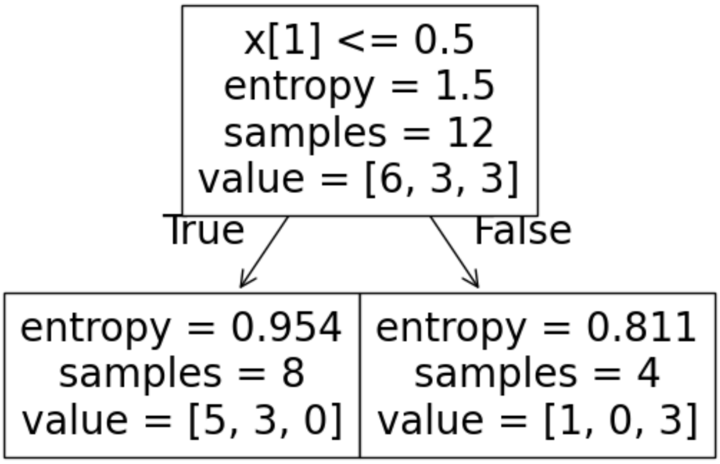
However if we are limited to a factor of 2, then case 1 (sunny, windy+rainy) is the best case scenario since it produces the highest information gain

3)

```
Names (ID): Huzaifa Mohammed (40242080), Mohammed Shurrab (40323793),
 Oleksandr Yasinovskyy (40241188)
*no limit on tree-depth*
Prediction:  ['Tennis']
Prediction Probability:  [[0. 0. 1.]]
```



```
* max_depth = 1 *
Predicition:  ['Tennis']
Prediction Probability:  [[0.25 0.   0.75]]
```



Question 6:

a)

Q6)

a) check if $h1(n)$ is consistent :-

$h1(n) \leq c(n,n') + h(n')$ ⇒ where n' is successor node of n

- check A

$A \rightarrow B$ (cost = 1) , $h1(A) = 10$, $h1(B) = 12$

⇒ $10 \leq 1 + 12 = 13$ ✓

$A \rightarrow C$ (cost = 4) , $h1(C) = 10$

⇒ $10 \leq 4 + 10 = 14$ ✓

$B \rightarrow C$ (cost = 1)

⇒ $12 \leq 1 + 10 = 1$ ✗

$B \rightarrow D$ (cost = 5) , $h1(D) = 8$

⇒ $12 \leq 5 + 8 = 13$ ✓

$C \rightarrow D$ (cost = 3)

$10 \leq 3 + 8 = 11$ ✓

$D \rightarrow E$ (cost = 8) , $h1(E) = 1$

$8 \leq 8 + 1 = 9$ ✓

$D \rightarrow G$ (cost = 9) , $h1(G) = 0$

$8 \leq 9 + 0 = 9$ ✓

$D \rightarrow F$ (cost = 3) , $h1(F) = 4.5$

$8 \leq 3 + 4.5 = 7.5$ ✗

∴ $h1(n)$ is $\boxed{\text{not consistent}}$ for two reasons

① All the edges in the graph are bidirectional & since $h1(n)$ is not optimal $h1(n) \neq c(n,n') + h(n')$ for all nodes in the graph

② There exists atleast 1 edge that does not satisfy $h1(n) \leq c(n,n') + h1(n')$

b)

b) run Algorithm A:

Step	visited	closed	open
1	–	–	A(10)
2	A	A	B(1+12 = 13), C(4+10 = 14)
3	B	A,B	C(1+1+10=12) , D(1+5+8 = 14) ⇒ C(12) from B
4	C	A,B,C	D(1+1+3+8 = 13) ⇒ D(13) from C
5	D	A,B,C,D	E(1+1+3+8+1 = 14), F(1+1+3+3+4.5 = 12.5) , G(1+1+3+9+0 = 13)
6	F	A,B,C,D,F	E(14) , G(1+1+3+3+5+0 = 13)
7	G	A,B,C,D,F,G	E(14)

The found path: $\boxed{A \rightarrow B \rightarrow C \rightarrow D \rightarrow F \rightarrow G}$

c)

c) range of values for $h_2(B)$ to make $h_2(n)$ admissible.

$h_2(B) \leq$ actual cost of optimal path from $B \to G$

optimal path from B

$B \to C(1) \to D(3) \to F(3) \to G(5)$

total cost $= 1+3+3+5 = 12$

$\boxed{h_2(B) \leq 12}$

d)

d) range of values to make $h_2(n)$ consistent:

$$h_2(n) \leq c(n, n') + h(n')$$

$B \to A$

$h_2(B) \leq 1 + 10 = 11$

$B \to C$

$h_2(B) \leq 1 + 9 = 10$

$B \to D$

$h_2(B) \leq 5 + 7 = 12$

$A \to B$

$h_2(A) \leq 1 + h_2(B)$

$h_2(B) \geq 10 - 1 = 9$

$C \to B$

$h_2(B) \geq 9 - 1 = 8$

$D \to B$

$h_2(B) \geq 7 - 5 = 2$

$\boxed{\text{Hence } 9 \leq h_2(B) \leq 10}$ to make $h_2(n)$ consistent

Question 7:

a) State can be represented this way:

```
board = [
['X', '-', '-'],
['-', '-', '-'],
['-', '-', '-']
]
Player = 'MIN'
Allowed_moves = 2
```

b) Max (X) moves once:

X - - - - - - - -	- X - - - - - - -	- - - - X - - - -

Max (X) moves twice:

X X - - - - - - -	X - X - - - - - -	X - - - X - - - -
X - - - - X - - -	- X - - - - - X -	- X - X - - - - -
- X - - X - - - -		

c) Total branching factor is 3 + 7 = 10.

Question 8:

a) State space:

Each state can be represented as a pair (E, F) where E is the remaining portion of English text to be translated, and F is the current sequence of French words generated so far. The full state space would be the set of all possible partial translations from English

b) Initial state:

(E_0, F_0) where E_0 is the full English sentence and F_0 is the empty French sequence since no words would be translated yet

c) Action:

Each action represents selection a French word or phrase that is a valid translation for the English word or phrase.

d) Transition model:

Given state (E,F) and action (choosing a French word/phrase for the English token), remove translated portion from E and append the French portion into F. The result is a new state (E', F') where E' is the remaining English sentence and F' is the extended French translation that we currently have.

e) Goal test:

The goal is reached when the entire English sentence has been consumed/translated (E_f is empty) and the French sentence F_f is a complete and grammatically correct translation

f) Path cost:

Language model probability: score how fluent or likely the French translation is where P (E|F)

Translation probability: how likely the French translation F is a valid translation of E.

Cost will be: Cost (E|F) = -*log* P(F|E)

Question 9:

a)

```
## Step 2: Load Dataset
df = pd.read_excel("Wallet.xlsx")
print(df.head())

## Step 3: Preprocess Data
X = df.drop("wallet", axis=1)
y = df["wallet"] - 1  # convert to 0,1,2

# Split data
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25, random_state=6721)
```

✓ 0.0s

```
   wallet  male  business  punish  explain
0       2     0         0       2        0
1       2     0         0       2        1
2       3     0         0       1        1
3       3     0         0       2        0
4       1     1         0       1        1
```

b)

```
model = XGBClassifier(
    learning_rate=0.1,
    max_depth=2,
    n_estimators=100,
    objective="multi:softmax",
    num_class=3,
    eval_metric="mlogloss",
    use_label_encoder=False
)
```

```
--- Training Set ---
Accuracy: 0.678
Precision (macro): 0.702
Recall (macro): 0.519
F1 Score (macro): 0.514
Confusion Matrix:
[[10  0 10]
 [ 2  4 30]
 [ 4  1 85]]
Classification Report:
              precision    recall  f1-score   support

           0       0.62      0.50      0.56        20
           1       0.80      0.11      0.20        36
           2       0.68      0.94      0.79        90

    accuracy                           0.68       146
   macro avg       0.70      0.52      0.51       146
weighted avg       0.70      0.68      0.61       146
```

```
--- Test Set ---
Accuracy: 0.571
Precision (macro): 0.286
Recall (macro): 0.374
F1 Score (macro): 0.324
Confusion Matrix:
[[ 1  0  3]
 [ 3  0 11]
 [ 1  3 27]]
Classification Report:
              precision    recall  f1-score   support

           0       0.20      0.25      0.22         4
           1       0.00      0.00      0.00        14
           2       0.66      0.87      0.75        31

    accuracy                           0.57        49
   macro avg       0.29      0.37      0.32        49
weighted avg       0.43      0.57      0.49        49
```

c)

```
Comparison of Different Learning Rates and Tree Depths:
   Learning Rate  Max Depth  Test - Accuracy  Test - F1 Score  Train - Accuracy  Train - F1 Score
0          0.100          2            0.571            0.324             0.678             0.514
1          0.100          5            0.551            0.318             0.692             0.543
2          0.500          2            0.571            0.324             0.678             0.525
3          0.500          5            0.551            0.318             0.692             0.580

Names (ID):
Huzaifa Mohammed (40242080),
Mohammed Shurrab (40323793),
Oleksandr Yasinovskyy (40241188)
```

It is seen that the learning rate does not affect the testing dataset since Accuracy and F1-score do not change as the learning rate varies. Similarly, the training accuracy does not change as well, however, the F1-score of the training dataset slightly increases as learning rate increases, which may indicate a faster training (fitting) time.

For the max depth: The training dataset accuracy and F1-score both improves when the max tree depth is increased, however, the testing dataset accuracy and F1-score decrease, indicating that the model is more prone to **overfitting** as we increase the max depth.