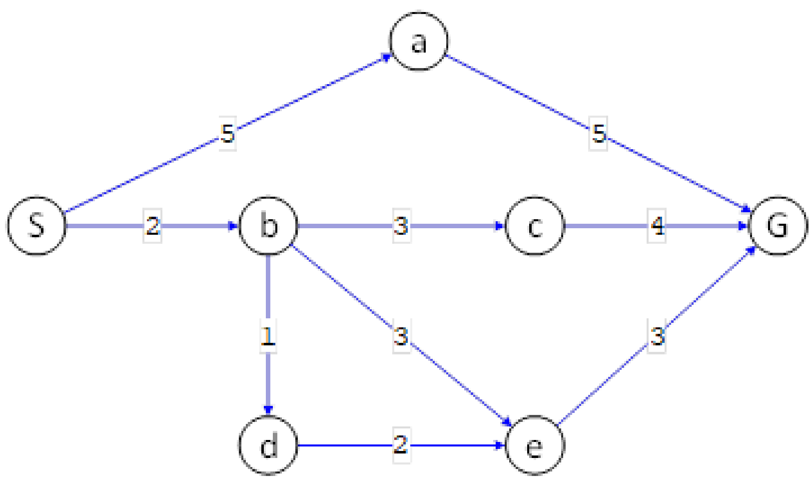
PART I: Obligatory Section (8pts)

You have to complete ALL of the following questions to get full credits of Part I.

Question 1 (2pts) Consider the following search problem and three heuristic functions h_1, h_2, h_3, where S is the start state and G is the goal state. Ties are broken in alphabetical order.



State	h_1	h_2	h_3
S	0	7	8
a	0	3	4
b	0	4	6
c	0	3	4
d	0	3	5
e	0	1	3
G	0	0	0

a) (1pt) Which of the heuristics shown above are admissible? Explain for every heuristic.

....

....

....

....

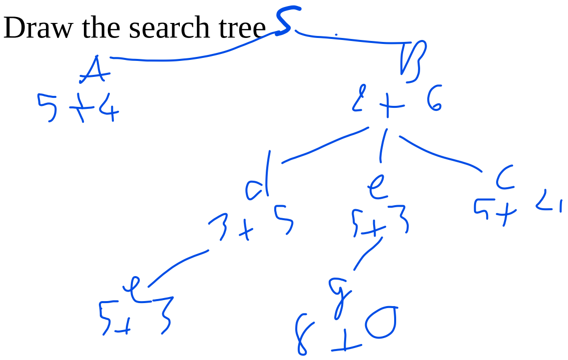
....

....

....

b) (1pt) Run A* algorithm using the heuristic h_3

- Draw the search tree



- Give the path the A* algorithm will return:

  S, B, E, G

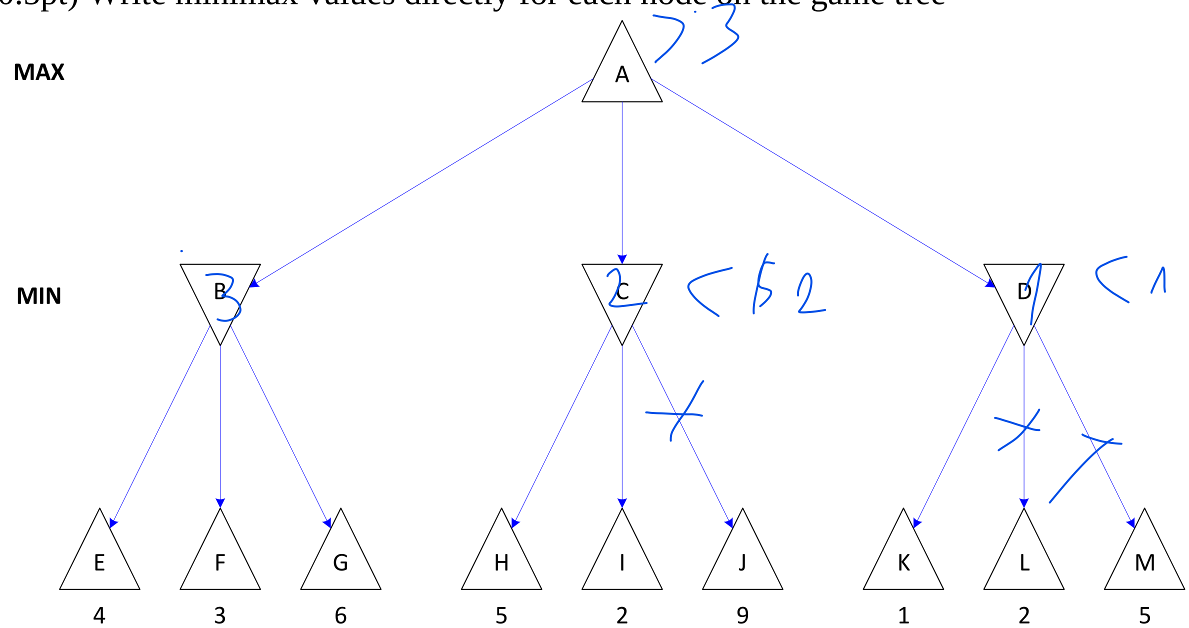
- Is this the shortest path from S to G? Explain your answer.

  Yes

  tree search A* and h3 is addmisible

Question 2 (1pt) Run the alpha-beta algorithm on the following two-player game tree to calculate the utility value for node A (assuming nodes are evaluated in left-to-right order).

a. (0.5pt) Write minimax values directly for each node on the game tree



b. (0.5pt) Will any nodes be pruned? Explain?

Question 3 (3pts) Knowledge representation

a) (1.5pts) Consider the following text. *"Every student takes either Databases or Artificial Intelligence. Every student who takes Artificial Intelligence knows Python. John is a student who did not take Databases".*

i. Build a FOL knowledge base from the text above, using only the given predicates

STUDENT(x): x is a student   TAKES(x, y): x takes y   KNOWS(x, y): x knows y

$\forall x$ Student(x) => Takes(x, D) $\vee$ Take(x, A)

$\forall x$ Student(x) $\cap$ Take(x, A) => Knows(x, P)

Student(John) $\cap$ $\neg$Takes(John, D)

ii. Prove via resolution that "*John knows Python.*"

1) $\neg$Student(x) $\vee$ Takes(x, D) $\vee$ Take(x, A)

2) $\neg$Student(x) $\vee$ $\neg$Take(x, A) $\vee$ Knows(x, P)

3) a) Student(J)

b) $\neg$Takes(J, D)

4) $\neg$Knows(J, P)

5) Takes(J, D) $\vee$ Take(J, A) (3a, 1) {x/J}

6) Take(J, A) (5, 3b)

7) Knows(J, P) (6, 3a, 2) {x/J}

8) Ø (7, 4)

Rewrite the given clauses in an appropriate form

Write down new clauses generated during the proof. For each new clause, state the source clauses required and the corresponding substitutions.

b) (0.5pt) Use the predicates given in a) to convert the following English sentences into FOL clauses. Consider brackets if you are not sure of the operator precedence.

i. Every student takes either Databases or Artificial Intelligence (but not both).

$\forall x \ Student(x) \Rightarrow [Takes(x, D) \Leftrightarrow \neg Takes(x, A)]$

ii. There is something that some student knows but the other students do not know.

$\exists x, y \ Student(x) \wedge Know(x, y)$
$[\forall z \ Student(z) \wedge (x \neq z) \Rightarrow \neg Knows(z, y)]$

c) (0.5pt) Is the below FOL sentence valid, satisfiable, or unsatisfiable? Explain your answer.

$$\underbrace{(\exists x \ x = x)}_{1} \rightarrow \underbrace{(\forall y \exists z \ y = z)}_{2}$$

1 is always True, 2 is not always True, 2, 1 is not related
$\Rightarrow$ Valid, not satisfiable

d) (0.5pt) For each of the following pairs of FOL clauses, present the MGU if there is any, otherwise explain why. Note that A is a constant.

i. $f(\ x, g(f(A), u)\ )$ and $f(\ g(u, v), x\ )$

....

....

ii. $p(\ A, x, f(g(y))\ )$ and $p(\ z, f(z), f(A))$

....

....

Question 4 (2pts) Using the decision tree learning algorithm (ID3), draw the decision tree for the following dataset while showing all step-by-step calculations. Knowing that the best attributes are chosen following the Information Gain measure.

ID	Age <30	Eat Pizza	Exercise	Result
1	Yes	Yes	Yes	Fit
2	Yes	Yes	No	Fit
3	Yes	No	Yes	Fit
4	Yes	No	No	Fit
5	No	Yes	Yes	Unfit
6	No	Yes	No	Unfit
7	No	No	Yes	Fit
8	No	No	No	Unfit

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

PART II: Optional Section (2pts)

You have to complete at least ONE of the two following questions to get full credits of Part II. The remaining question will be left for bonus credits.

Question 5 (2pts) Consider an Artificial Neural Network which has been trained to learn the following rule to categorize the brightness of a 2×2 black and white pixel images:

- If it contains 3 or 4 black pixels, it is *DARK*;
- If it contains 0, 1 or 2 black pixels, it is *BRIGHT*.

We can model this with a **perceptron** by saying that there are 4 input units, one for each pixel (+1 if the pixel is white and –1 if the pixel is black). The output unit produces +1 if the input image is to be categorized as *BRIGHT* and –1 if the image is *DARK*.

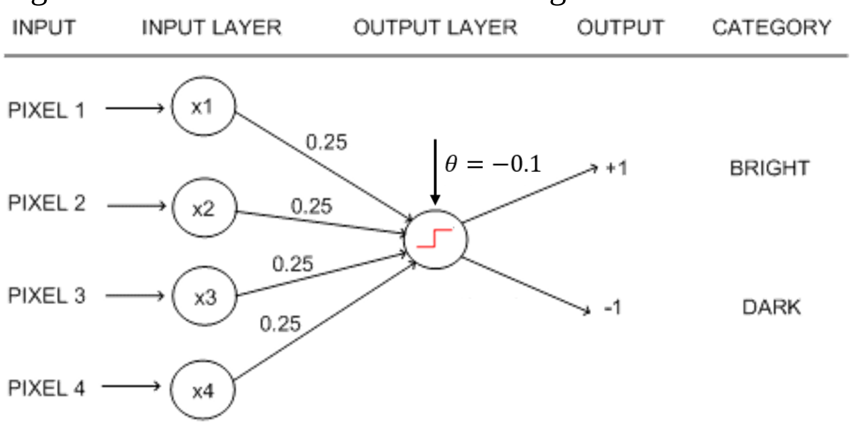


Figure 1. The perceptron for Question 5.

If we choose the weights as shown in Figure 1, can the perceptron perfectly categorize all images of four pixels into *DARK* and *BRIGHT*? Explain your answer.

Knowing that the output is calculated by the following equation:

$$Y = \text{sign}\left[\sum_{i=1}^{4} x_i w_i - \theta\right]$$

where $\theta = -0.1$ and $\text{sign}(t) = \begin{cases} +1 & if\ t \geq 0 \\ -1 & if\ t < 0 \end{cases}$

Question 6 (2pts) Consider the cryptarithmetic problem shown aside. Each letter stands for a distinct digit. The aim is to find a substitution of digits for letters such that the resulting sum is arithmetically correct, with the added restriction that no leading zeroes are allowed.

$$\begin{array}{r} O\,N\,E \\ +\;\; O\,N\,E \\ \hline T\,W\,O \end{array}$$

Figure 2 is the constraint hypergraph for the problem, showing the Alldiff constraint (rectangle box at the top) as well as the column addition constraints (three square boxes in the middle). The variables C_1 and C_2 represent the carry digits for the two columns, and thus $C_1, C_2 \in \{0, 1\}$.

Ties of MRV are broken in alphabetical order. Ties of LCV are broken by taking the smaller values (from 1 to 9)

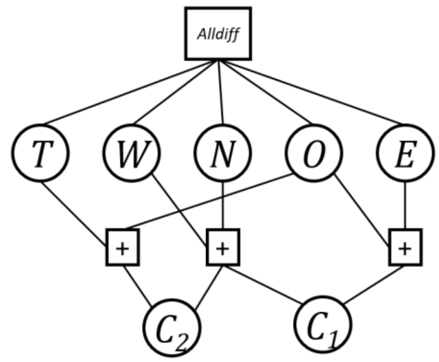


Figure 2. The constraint hypergraph for Question 6.

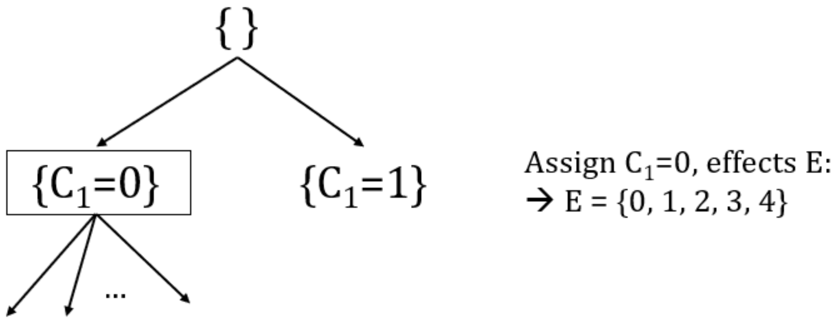
a) (1pt) Formulate this problem as a constraint satisfaction problem. That is, identify the variables, unary/binary constraints, and the domains of the variables.

b) (1pt) Solve the problem using the strategy of backtracking with forward checking and the MRV and least-constraining-value heuristics.

- Draw the search tree of the expanding nodes? Only consider assignments to a single variable at each node. States are defined by the values assigned so far
    - Initial state: an empty assignment {}
    - Goal state: the current assignment is complete.

At each step, show the domain of the variables affected by forward checking. (0.5pt) Some first steps are given below.

- What is the first returned solution? (0.5pt)



--THE END--