EXAM 23T2 FCruz

Note for current students: This exam is an old version taken online in a similar course. It is given to you just to study the concepts and not as an example of the coming exam in our course.

To be consider before starting:

- The time to complete the exam is 2 hours.
- The exam contains 13 multiple-choice questions.
- 11 questions give 4 marks. Only 2 questions give 8 marks.
- Total marks are 60.

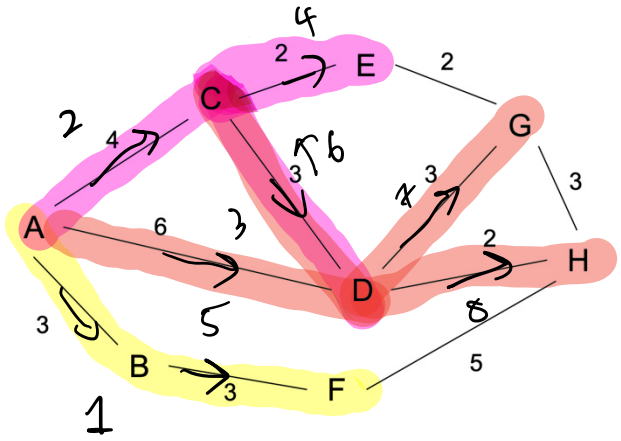
The summary of questions is as follows:

Topic	Question	Marks
Search	IDDFS	4
Search	A*	4
Reinforcement learning	Softmax	4
Reinforcement learning	Returns	4
Neural networks	Single-layer perceptron	4
Neural networks	Neural design	4
Optimization	Tabu search	8
Computer vision	Averaging method	4
Computer vision	Split-and-merge method	4
Language processing	Bigram probability	4
Language processing	Minimum edit distance	4
Uncertain reasoning	Bayes nets	8
Uncertain reasoning	Fuzzy logic	4

TOPIC: Search – Question: IDDFS [4 marks]

Consider the following road map with distances indicated on lines drawn between towns (the map is not to scale). The straight-line distances form each town to H are listed in the table.

depth 2 iteration



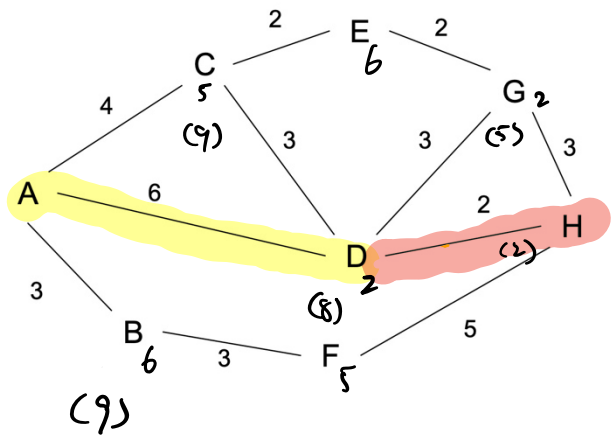
Town	Distance to H
A	7
B	6
C	5
D	2
E	6
F	5
G	2
H	0

What order are nodes expanded by iterative deepening depth first search when searching for a path between A and H? Where there is choice of nodes, take the first one by alphabetical ordering. Assume the search algorithm includes c*ycle checking* along a path, Tree-Search-IDDFS. **Stop the search once the goal node is expanded**.

a. A ABCD ABFCDEDCGH
b. A ABCD ABFCDH
c. A ABCD ABCDE ABFCDH
d. A ABCD ABFH
e. A ABCD ABFCDEGH

TOPIC: Search – Question: A* [4 marks]

Consider the following road map with distances indicated on lines drawn between towns (the map is not to scale). The straight-line distances form each town to H are listed in the table.



Town	Distance to H
A	7
B	6
C	5
D	2
E	6
F	5
G	2
H	0

What order are nodes expanded by A* search using the straight-line distances to H in the table as the heuristic function, when searching for a path between A and H? Where there is choice of nodes, take the first one by alphabetical ordering. **Stop the search once the goal node is expanded**.

a. ADH
b. ABFH
c. ABCDEFGH
d. ACDH
e. ACEGH

TOPIC: Reinforcement learning – Question: Softmax [4 marks]

Consider an RL agent navigating a gridworld, with four possible action: up (U), down (D), left (L), and right (R). The agent uses the softmax action selection method. Remember this method computes the probability of selecting an action using a Boltzmann distribution, as follows:

$$P(s_t, a) = \frac{e^{Q(s_t,a)/T}}{\sum_{a_i \in A} e^{Q(s_t,a_i)/T}}$$

In a particular given state S_t, the agent has the following Q-values to decide what action to take next:

Q(U)	Q(D)	Q(L)	Q(R)
0.7698	0.6501	0.0252	-0.7698

What would be the action selected by the agent if the temperature T used is 0.9 and the random number drawn is 0.9021.

Answer:

a. Up
b. Down
c. Left
d. Right
e. The agent would stay at the same position

Explanation (do not use in Moodle):

P(U) = 0.401054937, P(D) = 0.351109598, P(L) = 0.175346688, P(R) = 0.072488777

Accumulated P = {0.401054937, 0.752164535, 0.927511223, 1}

0.7021 falls in the range (0.75, 0.927)

TOPIC: Reinforcement learning – Question: Returns [4 marks]

Consider the return equation shown below with a discount factor γ = 0.9 and a reward sequence of 7, 3, 1, 10, -10. The return G_0 is equal to:

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + ... = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}$$

Answer:

$G_0 = R_1 + \gamma R_2 + \gamma^2 R_3 + \gamma^3 R_4 + \gamma^4 R_5$

a. 4.71
b. 1
c. 1.9
d. 11.239
e. 9.2

Explanation (do not use in Moodle):

G_5 = 0 (and all the following ones)

G_4 = -10

G_3 = 10 + 0.9 x G4 = 10 – 9 = 1

G_2 = 1 + 0.9 x G3 = 1 + 0.9 = 1.9

G_1 = 3 + 0.9 x G2 = 3 + 1.71 = 4.71

G_0 = 7 + 0.9 x G1 = 7 + 4.239= = 11.239

TOPIC: Neural networks – Question: Single-layer perceptron [4 marks]

Consider the training data shown in the following Table to divide the space. Using the single-layer perceptron learning rule with a learning rate $\alpha = 1.0$ and initial weights $w_1 = 1$, $w_2 = 0$, and $b = 1.5$, what would be the final value of the weight after convergence?

Training example	x_1	x_2	Class
a	0.0	0.0	1
b	1.0	2.0	-1
c	2.0	-1.0	-1
d	-2.0	1.0	1

Answer:

inter 1. $y = \text{sign}(0.1 + 0.0 + 1.5) = \text{sign}(1.6) = 1$

$w_1 = 1 + 0.1 \times (1 - 1) \times 0.0 = 1.0$

a. $w_1 = 1.0$, $w_2 = 0.0$, $b = 1.5$
b. $w_1 = -5.0$, $w_2 = -2.0$, $b = 1.5$
c. $w_1 = -5.0$, $w_2 = -2.0$, $b = -0.5$
d. $w_1 = -5.0$, $w_2 = -2.0$, $b = -2.5$
e. $w_1 = -1.0$, $w_2 = -4.0$, $b = -0.5$

Explanation (do not use in Moodle):

Iteration	bias	w1	w2	Example	x1	x2	Class	Prediction	Rounded	Action	Delta/X
1	1.5	1.0	0.0	a	0.0	0.0	1	1.5	1	NONE	0.0
2	1.5	1.0	0.0	b	1.0	2.0	-1	2.5	1	SUBSTRACT	-2.0
3	-0.5	-1.0	-4.0	c	2.0	-1.0	-1	1.5	1	SUBSTRACT	-2.0
4	-2.5	-5.0	-2.0	d	-2.0	1.0	1	5.5	1	NONE	0.0
5	-2.5	-5.0	-2.0	a	0.0	0.0	1	-2.5	-1	ADD	2.0
6	-0.5	-5.0	-2.0	b	1.0	2.0	-1	-9.5	-1	NONE	0.0
7	-0.5	-5.0	-2.0	c	2.0	-1.0	-1	-8.5	-1	NONE	0.0
8	-0.5	-5.0	-2.0	d	-2.0	1.0	1	7.5	1	NONE	0.0
9	-0.5	-5.0	-2.0	a	0.0	0.0	1	-0.5	-1	ADD	2.0
10	1.5	-5.0	-2.0	b	1.0	2.0	-1	-7.5	-1	NONE	0.0
11	1.5	-5.0	-2.0	c	2.0	-1.0	-1	-6.5	-1	NONE	0.0
12	1.5	-5.0	-2.0	d	-2.0	1.0	1	9.5	1	NONE	0.0
13	1.5	-5.0	-2.0	a	0.0	0.0	1	1.5	1	NONE	0.0
14	1.5	-5.0	-2.0	b	1.0	2.0	-1	-7.5	-1	NONE	0.0
15	1.5	-5.0	-2.0	c	2.0	-1.0	-1	-6.5	-1	NONE	0.0
16	1.5	-5.0	-2.0	d	-2.0	1.0	1	9.5	1	NONE	0.0

TOPIC: Neural networks – Question: Neural design [4 marks]

Taking into account the good practices for neural design, what would be the recommended number of neurons in the hidden layer for a multilayer perceptron to approximate a non-linear function of 6 inputs and 6 outputs if you have available 10,000 samples? Consider 60% data for training, 30% for test, and 10% for generalization.

Training = 10000 × 60% = 6000

Answer:

we want $N_w < 600$

$N_w = (N_i + 1) \times N_h + (N_h + 1) \times N_o$

a. 54
b. 76
c. 68
d. 45
e. 53

Explanation (do not use in Moodle):

What it is known from the previous description is the following:

Ni = 6
No = 6
$N_{samples}$ = 6,000 as we use only 60% for training.
Nh =?

But we also know that Nw = (Ni + 1) × Nh + (Nh + 1) × No and Nw < $N_{samples}$/10, then:

600 = (6 + 1) × Nh + (Nh + 1) × 6
600 = 7Nh + 6Nh + 6
594 = 13Nh
Nh = 594/13

Nh = 45.69 → 45 as the number of neurons in the hidden layer is an integer. It might be 46, however, with 45 we keep true the relationship Nw < $N_{samples}$/10.

TOPIC: Optimization – Question: Tabu Search [8 marks]

Consider an optimization problem and a metaheuristic to find an approximate solution. The fitness function to evaluate candidate solutions is defined as $y = x^2$. You are asked to implement the Tabu Search algorithm shown in the code below, taking into account the following:

- For the initial solution, use a random number approximately between -10 and 10 from a standard normal distribution (i.e., μ=0, σ=1).
- For the Tabu List use a numpy array.
- The neighbourhood size is equal to 50 for each iteration.
- Each neighbour is computed based on the current best solution as: best solution + a random number from a standard normal distribution.
- The termination criterion is to find a solution with fitness lower than 1e-10.
- Use a seed for random numbers equal to 55.

What would be the returned solution by the algorithm??

Algorithm 1 Tabu search optimisation method.

$s_0 \leftarrow$ generate initial solution
$s_{best} \leftarrow s_0$
tabuList $\leftarrow \{s_0\}$
repeat
  $\{s_1, s_2, ..., s_n\} \leftarrow$ generate neighbourhood from (s_{best})
  $s_{candidate} \leftarrow s_1$
  for i $\leftarrow$ 2 TO n **do**
    $\delta \leftarrow f(s_i) - f(s_{candidate})$
    if s_i is not in tabuList **and** $\delta < 0$ **then**
      $s_{candidate} \leftarrow s_i$
    end if
  end for
  $s_{best} \leftarrow s_{candidate}$
  Add $s_{candidate}$ to tabuList
until a termination criterion is satisfied
return s_{best}

Answer:

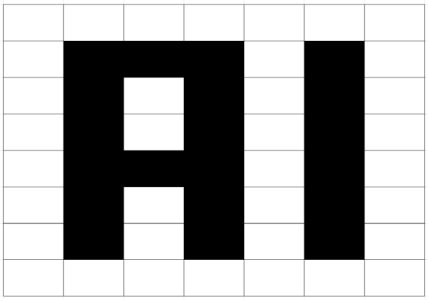
a. -2.0387e-06
b. -1.6237e-07
c. -1.5683e-06
d. 8.1281e-07
e. 1.0815e-06

Explanation (do not use in Moodle):

```python
# -*- coding: utf-8 -*-
"""
Created on Mon Aug  7 21:37:08 2023

@author: Francisco
"""

import numpy as np

def fitness(x):
    return (x)**2.0

def tabuSearch(neighbourhoodSize):
    initialSolution = np.random.randn() * 10
    bestSolution = initialSolution
    print(initialSolution)

    tabuList = np.array([])
    tabuList = np.append(tabuList, initialSolution)

    while True:
        neighbours = np.zeros(neighbourhoodSize)
        for j in range(neighbourhoodSize):
            neighbours[j] = bestSolution + np.random.randn()

        candidateSolution = neighbours[0]
        for i in range(1, neighbourhoodSize):
            delta = fitness(neighbours[i]) - fitness(candidateSolution)
            if (neighbours[i] not in tabuList) and (delta < 0):
                candidateSolution = neighbours[i]

        bestSolution = candidateSolution
        tabuList = np.append(tabuList, candidateSolution)

        if fitness(bestSolution) < 1e-10:
            break

    return bestSolution

np.random.seed(55)
neighbourhoodSize = 50
bestSolution = tabuSearch(neighbourhoodSize)
print('Returned solution: f(%s) = %f' % (bestSolution, fitness(bestSolution)))

```

TOPIC: Computer vision – Question: Averaging method [4 marks]

Consider the binary image with dimension 8x7 shown below. The image can be represented in a 2-dimensional array using 1's and 0's as shown in the table below. Using the averaging method (creating a new image) with a threshold ε = 2 and a 3x3 sliding window, what would be the resulting image?



Sum = 3 > 2

= 1

0	0	0	0	0	0	0
0	**1**	**1**	**1**	0	**1**	0
0	**1**	0	**1**	0	**1**	0
0	**1**	0	**1**	0	**1**	0
0	**1**	**1**	**1**	0	**1**	0
0	**1**	0	**1**	0	**1**	0
0	**1**	0	**1**	0	**1**	0
0	0	0	0	0	0	0

Answer:

Image 1

0	0	0	0	0	0	0
0	**1**	**1**	**1**	**1**	**1**	0
0	**1**	**1**	**1**	**1**	**1**	0
0	**1**	**1**	**1**	**1**	**1**	0
0	**1**	**1**	**1**	**1**	**1**	0
0	**1**	**1**	**1**	**1**	**1**	0
0	**1**	0	**1**	0	**1**	0
0	0	0	0	0	0	0

Image 2

0	0	0	0	0	0	0
0	**1**	**1**	**1**	**1**	**1**	0
0	**1**	**1**	**1**	**1**	**1**	0
0	**1**	**1**	**1**	**1**	**1**	0
0	**1**	**1**	**1**	**1**	**1**	0
0	**1**	**1**	**1**	**1**	**1**	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

Image 3

0	0	0	0	0	0	0
0	**1**	**1**	**1**	**1**	0	0
0	**1**	**1**	**1**	**1**	**1**	0
0	**1**	**1**	**1**	**1**	**1**	0
0	**1**	**1**	**1**	**1**	**1**	0
0	**1**	**1**	**1**	**1**	**1**	0
0	0	**1**	0	**1**	0	0
0	0	0	0	0	0	0

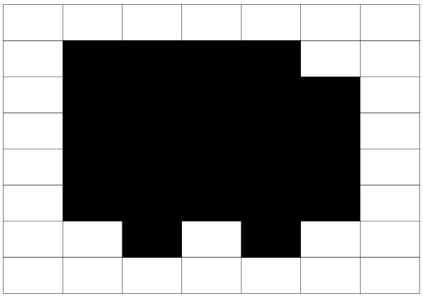
Image 4

0	0	0	0	0	0	0
0	**1**	**1**	**1**	0	**1**	0
0	**1**	0	**1**	0	**1**	0
0	**1**	0	**1**	0	**1**	0
0	**1**	**1**	**1**	0	**1**	0
0	**1**	0	**1**	0	**1**	0
0	**1**	0	**1**	0	**1**	0
0	0	0	0	0	0	0

a. Image 1
b. Image 2

c. Image 3
d. Image 4
e. All images are valid results for the averaging method

Explanation (do not use in Moodle):



TOPIC: Computer vision – Question: Split-and-merge method [4 marks]

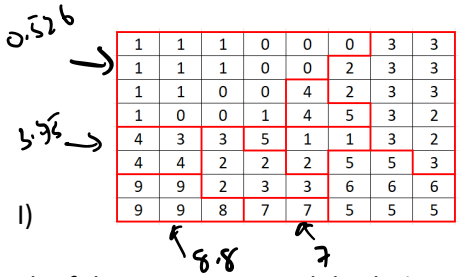
Consider the following 8x8 picture matrix.

1	1	1	0	0	0	3	3
1	1	1	0	0	2	3	3
1	1	0	0	4	2	3	3
1	0	0	1	4	5	3	2
4	3	3	5	1	1	3	2
4	4	2	2	2	5	5	3
9	9	2	3	3	6	6	6
9	9	8	7	7	5	5	5

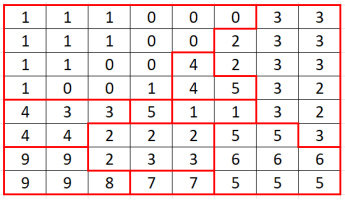
split if maximum difference of region > 1
merge if two regions average difference ≤ 1

Using the split-and-merge method for region finding, two possible solutions with ε <= 1 might be:

I)



II)



Which of the previous are valid solutions for the split-and-merge method with ε <= 1?

Answer:

a. The split-and-merge method cannot be used in this image
b. Just I
c. Just II
d. Both
e. None

TOPIC: Language processing – Question: Bigram probability [4 marks]

The bigram probability is computed as:

exact word match

$$P(w_i \mid w_{i-1}) = \frac{c(w_{i-1}, w_i)}{c(w_{i-1})}$$

Considering the following mini corpus containing 5 sentences:

<s> There is not much time until the next weekend </s>

<s> I love spending weekends at the beach </s>

<s> Next weekend I will go to the cinema </s>

<s> I went the last weekend at the stadium </s>

<s> Maybe I am thinking too much about the weekend </s>

What is the bigram probability P(at|weekend)?

$$\frac{c(\text{weekend, at})}{c(\text{weekend})}$$

Answer:

a. 1.0
b. 0.4
c. 0.25
d. 0.2
e. 0.8

Explanation (do not use in Moodle):
P(at|weekend) = 1/4 = 0.25

TOPIC: Language processing – Question: Minimum edit distance [4 marks]

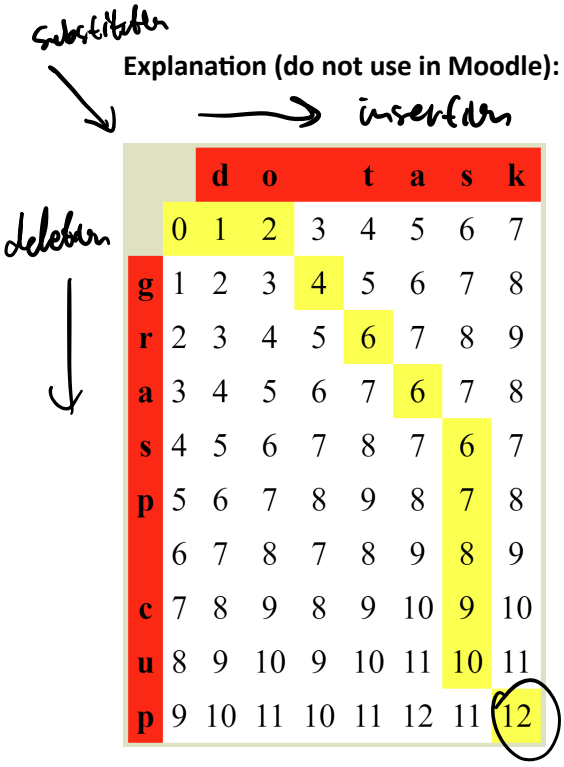
In a human-robot interaction scenario, a cleaning-table robot might be instructed by a human trainer with different advice, such as: get the cup, grasp cup, move left, move right, clean table, do task, among others. As the advice is given using an automatic speech recognition system, sometimes the received sentence might be wrongly identified.

We are particularly interested in knowing the minimum edit distance between two sentences: grasp cup and do task. Assuming insertion and deletion costs equal to 1 and substitution equal to 2, determine the Levenshtein distance for the two strings.

Answer:

a. 12
b. 8
c. 14
d. 10
e. 9

Explanation (do not use in Moodle):



		d	o		t	a	s	k
	0	1	2	3	4	5	6	7
g	1	2	3	4	5	6	7	8
r	2	3	4	5	6	7	8	9
a	3	4	5	6	7	6	7	8
s	4	5	6	7	8	7	6	7
p	5	6	7	8	9	8	7	8
	6	7	8	7	8	9	8	9
c	7	8	9	8	9	10	9	10
u	8	9	10	9	10	11	10	11
p	9	10	11	10	11	12	11	12

TOPIC: Uncertain reasoning – Question: Bayes network [8 marks]

Consider the following random variables:

rsp: regular sport practice.

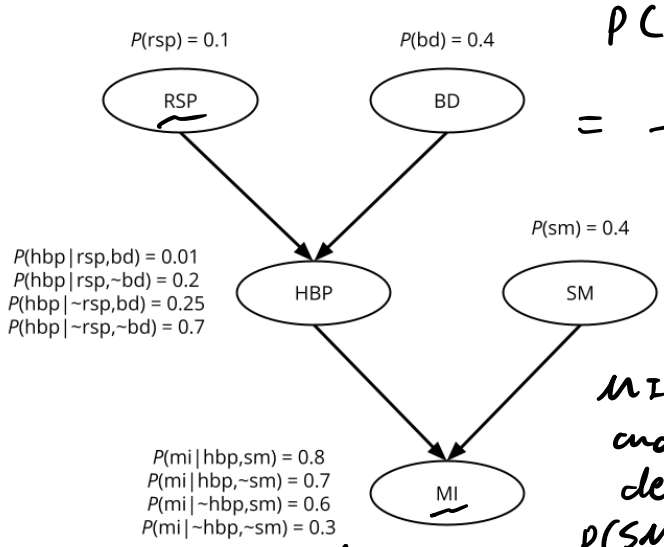
bd: balanced diet.

hbp: high blood pressure.

sm: smoker.

mi: has suffered a myocardial infarction.

The causal relationships and the probabilistic knowledge are shown in the Bayes network below. Using the network, compute the probability of being a smoker given that a myocardial infarction has been suffered and no regular sport is practised, i.e., P(SM|MI, ~RSP).



Answer:

a. 0.4853
b. 0.4
c. 0.6872
d. 0.7
e. 0.5147

Explanation (do not use in Moodle):

```
import pomegranate as pg

rsp = pg.DiscreteDistribution({'regular sport': 0.1, 'no regular sport': 0.9})
bd = pg.DiscreteDistribution({'balanced diet': 0.4, 'no balanced diet': 0.6})
sm = pg.DiscreteDistribution({'smoker': 0.4, 'no smoker': 0.6})

hbp = pg.ConditionalProbabilityTable(
        [[ 'regular sport', 'balanced diet', 'high blood pressure', 0.01 ],
         [ 'regular sport', 'no balanced diet', 'high blood pressure', 0.2 ],
         [ 'no regular sport', 'balanced diet', 'high blood pressure', 0.25 ],
         [ 'no regular sport', 'no balanced diet', 'high blood pressure', 0.7 ],
         [ 'regular sport', 'balanced diet', 'no high blood pressure', 0.99 ],
         [ 'regular sport', 'no balanced diet', 'no high blood pressure', 0.8 ],
         [ 'no regular sport', 'balanced diet', 'no high blood pressure', 0.75 ],
         [ 'no regular sport', 'no balanced diet', 'no high blood pressure', 0.3 ]],[rsp,bd])

mi = pg.ConditionalProbabilityTable(
        [[ 'high blood pressure', 'smoker', 'myocardial infarction', 0.8 ],
         [ 'high blood pressure', 'no smoker', 'myocardial infarction', 0.7 ],
         [ 'no high blood pressure', 'smoker', 'myocardial infarction', 0.6 ],
         [ 'no high blood pressure', 'no smoker', 'myocardial infarction', 0.3 ],
         [ 'high blood pressure', 'smoker', 'no myocardial infarction', 0.2 ],
         [ 'high blood pressure', 'no smoker', 'no myocardial infarction', 0.3 ],
         [ 'no high blood pressure', 'smoker', 'no myocardial infarction', 0.4 ],
         [ 'no high blood pressure', 'no smoker', 'no myocardial infarction', 0.7 ]],[hbp,sm])

model = pg.BayesianNetwork("MyBN")

RSP = pg.State(rsp, name="RSP")
BD = pg.State(bd, name="BD")
HBP = pg.State(hbp, name="HBP")
SM = pg.State(sm, name="SM")
MI = pg.State(mi, name="MI")

model.add_states(RSP, BD, HBP, SM, MI)
model.add_edge(RSP, HBP)
model.add_edge(BD, HBP)
model.add_edge(HBP, MI)
model.add_edge(SM, MI)

model.bake()
# P(SM|MI, ~RSP)
print(model.predict_proba([{'MI': 'myocardial infarction'}, {'RSP': 'no regular sport'}])[0][3].parameters[0]['smoker'])
```

TOPIC: Uncertain reasoning – Question: Fuzzy logic [4 marks]

In a streaming platform, the analysis team is performing a study on how a "normal customer" uses the platform. Based on the number of hours each customer uses the platform, they have ordered the customers and defined a fuzzy set for normal customers using a trapezoid membership function. The parameters for the membership function are [5, 20, 60, 95] which means:

- Customers in the lowest 5% of use have a membership equal to zero.
- Customers between 5% and 20% of use have an increasing membership from zero to one.
- Customers between 20% and 60% of use have a membership equal to one.
- Customers between 60% and 95% of use have a decreasing membership from one to zero.
- Customers in the highest 5% of use (between 95% and 100%) have a membership equal to zero.

For a customer located at position 72.5% of use compared with other customers, what would be the membership value?

$$\mu(x) = \frac{b-x}{b-a}$$

$$\mu(72.5) = \frac{95-72.5}{95-60} \approx 0.6428$$

Answer:

a. 0.6428
b. 0.5
c. 0.7142
d. 0.4285
e. 0.5714

Explanation (do not use in Moodle):

a	5
b	20
c	60
d	95
x	72.5
	4.5
	1
	0.642857
	0.642857
	0
	0.642857