

CONFIDENTIAL



UNIVERSITI TUN HUSSEIN ONN MALAYSIA

**FINAL EXAMINATION
SEMESTER I
SESSION 2019/2020**

COURSE NAME : ARTIFICIAL INTELLIGENCE
COURSE CODE : BIT 20903
PROGRAMME CODE : BIT
EXAMINATION DATE : DECEMBER 2019 / JANUARY 2020
DURATION : 3 HOURS
INSTRUCTION : ANSWER ALL QUESTIONS

TERBUKA

THIS QUESTION PAPER CONSISTS OF SIX (6) PAGES

CONFIDENTIAL

Intruccion: Answer ALL questions.

- Q1 (a) Define intelligence via Turing Test. (2 marks)
- (b) How Alan Turing proposed the test setup? (5 marks)
- (c) At the very minimum, to accomplish a human-comparable intelligence, the machine needs to be well versed with what? (8 marks)

Q2 (a) **Figure Q2(i)** and **Figure Q2(ii)** represent the input data and the boundaries of the input data obtained using a clustering method, respectively.

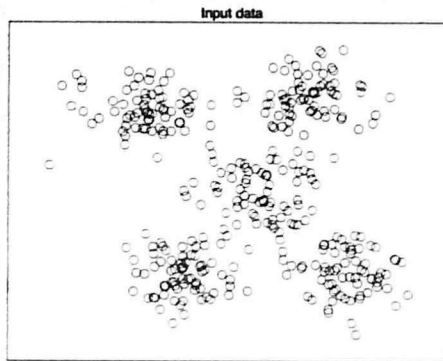


Figure Q2(i)

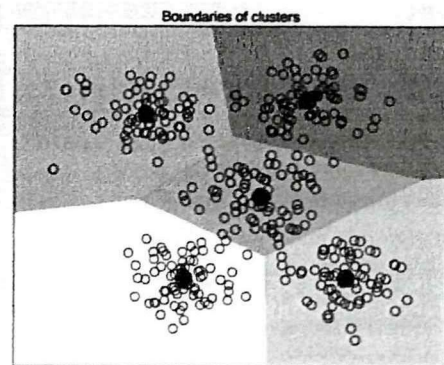


Figure Q2(ii)

What clustering method is being used for the input data above? And what is the black filled circle at the center of each cluster represents? (4 marks)

- (b) Nearest neighbors refers to the process of finding the closest points to the input point from the given dataset. When this method is frequently used? (2 marks)
- (c) Given a sample data and other related information below, after implementing the K Nearest Neighbors model (create, train and test in a Python environment), draw a **(ONE)** graph that represents the input data and a **(ONE)** graph that represents the k nearest neighbors. Please also include the test data point in the second graph and a sample of corresponding final output on your Terminal.

TERBUKA

```

# Input data
X = np.array ([[2.1, 1.3], [1.3, 3.2], [2.9, 2.5],
[2.7, 5.4], [3.8, 0.9], [7.3, 2.1], [4.2, 6.5],
[3.8, 3.7], [2.5, 4.1], [3.4, 1.9], [5.7, 3.5],
[6.1, 4.3], [5.1, 2.2], [6.2, 1.1]])

# Number of nearest neighbors
k = 5

# Test datapoint
test_datapoint = [4.3, 2.7]

```

(19 marks)

- Q3** (a) We encounter mathematical operations all the time. Logic programming is a very efficient way of comparing expressions and finding out unknown values. Based on the scenario of validating primes below, determine the Terminal output results.

```

import itertools as it
import logpy.core as lc
from sympy.ntheory.generate import prime, isprime

# Check if the elements of x are prime
def check_prime(x):
    if lc.isvar(x):
        return lc.condeseq([(lc.eq, x, p)] for p in
map(prime, it.count(1)))
    else:
        return lc.success if isprime(x) else lc.fail

# Declate the variable
x = lc.var()

# Check if an element in the list is a prime number
list_nums = (23, 4, 27, 17, 13, 10, 21, 29, 3, 32, 11,
19)
print('\nList of primes in the list:')
print(set(lc.run(0, x, (lc.membero, x, list_nums),
(check_prime, x))))

# Print first 7 prime numbers
print('\nList of first 7 prime numbers:')
print(lc.run(7, x, check_prime(x)))

```

(4 marks)



CONFIDENTIAL

- (b) Consider the following family tree in **Figure Q3** below:

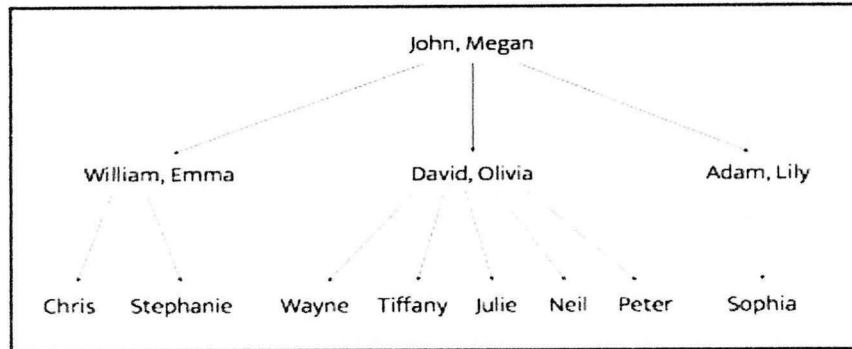


Figure Q3

List out all the possible results displayed on your Terminal based on the Python file below:

```

# Adam's parents
name = 'Adam'
output = run(0, x, parent(x, name))
print("\nList of " + name + "'s parents:")
for item in output:
    print(item)

# Wayne's grandparents
name = 'Wayne'
output = run(0, x, grandparent(x, name))
print("\nList of " + name + "'s grandparents:")
for item in output:
    print(item)

# Megan's grandchildren
name = 'Megan'
output = run(0, x, grandparent(name, x))
print("\nList of " + name + "'s grandchildren:")
for item in output:
    print(item)

# David's siblings
name = 'David'
output = run(0, x, sibling(x, name))
siblings = [x for x in output if x != name]
print("\nList of " + name + "'s siblings:")
for item in siblings:
    print(item)
  
```

TERBUKA

CONFIDENTIAL

```
# Tiffany's uncles
name = 'Tiffany'
name_father = run(0, x, father(x, name))[0]
output = run(0, x, uncle(x, name))
output = [x for x in output if x != name_father]
print("\nList of " + name + "'s uncles:")
for item in output:
    print(item)

# All spouses
a, b, c = var(), var(), var()
output = run(0, (a, b), (father, a, c), (mother, b, c))
print("\nList of all spouses:")
for item in output:
    print('Husband:', item[0], '<==> Wife:', item[1])
```

(16 marks)

- Q4** (a) Define heuristic search. (4 marks)
- (b) What is uninformed search? List down **3 (THREE)** examples of popular uninformed search techniques and how are they (uninformed search techniques) different from informed search technique? (8 marks)
- (c) We have a few regions in **Figure Q4** that are labeled with names. Our goal is to color it with four colors so that no adjacent regions within the figure have the same color by using the Constraint Satisfaction framework (one of the Heuristic Search technique).

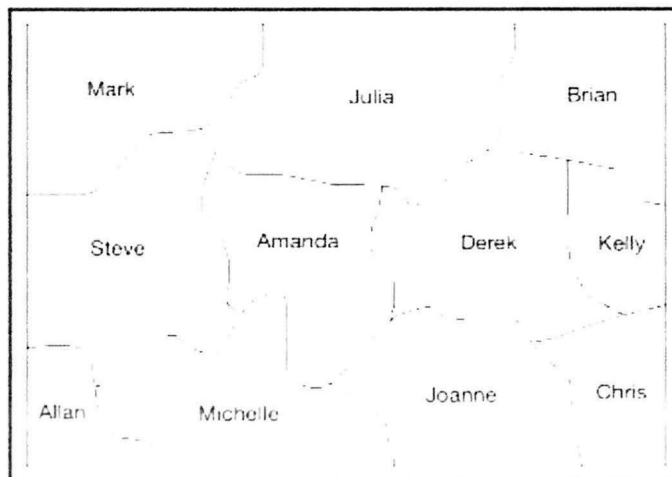


Figure Q4

TERBUKA

Define the constraint (pseudocode) that specifies that the value should be different in your Python program.

(2 marks)

- (d) Based on question **Q4(c)** above, list down the potential output color mapping for each of the labeled names on your Terminal. Consider that only 4 colors of 'red', 'green', 'blue' and 'gray' are to be used.

(6 marks)

- Q5** (a) Define perceptron with respect to the artificial neural network.

(4 marks)

- (b) How a neural network is trained?

(6 marks)

- (c) What type of layers are commonly used to construct a Convolutional Neural Network (CNN)?

(10 marks)

- END OF QUESTION -

TERBUKA