



OLLSCOIL NA GAILLIMHÉ
UNIVERSITY OF GALWAY

Semester I Examinations 2022-2023

Course Instance Code(s) 1MAI1, 1MAO3
Exam(s) MSc in Computer Science (Artificial Intelligence), MSc in
Computer Science (Artificial Intelligence) - Online
Module Code(s) CT5132, CT5148
Module(s) **Programming and Tools for Artificial Intelligence**
Discipline School of Computer Science

Paper No. 1

External Examiner Dr. John Woodward
Internal Examiner(s) Prof. Michael Madden
Dr. James McDermott *

Programme Coordinator(s) Dr. Matthias Nickles, Dr. James McDermott

Instructions **Answer all 4 questions. All are worth equal marks.**

Duration 2 Hours exam

Number of pages 5 (including this page)

Discipline Computer Science

Requirements

Release in Exam Venue Yes ☒ No ☐

Release to Library Yes ☒ No ☐

Calculator Allowed (non-programmable)

Question 1: Basic Python

- (a) Define a function named `palindrome` which accepts one string argument and tests whether it is palindromic (i.e. is the same whether read left-to-right or right-to-left) and returns a `bool`. Show how to call this function. [5]

- (b) Given the following data structure: [5]

```
x = [{'a': 1, 'b': 2}, {'a': 11, 'b': 12}, {'c': 100}]
```

Write an expression using `x` that will have the value 100.

- (c) For the following function, write a docstring containing three doctests to demonstrate usage. State where the docstring should be placed. [5]

```
def validate(s):
    if len(s) != 8:
        return 'Bad length'
    if not s[0].isupper():
        return 'Initial uppercase required'
    return 'Ok'
```

- (d) Suppose we have a dict, `d`. How can we create a new dict in which the keys are sorted alphabetically? You can answer with code or in (precise) English. [5]

- (e) In the following code, explain what happens when we call `len(c)`. [5]

```
class C:
    def __init__(self, data):
        self.data = data
    def __str__(self):
        return str(self.data)
    def __len__(self):
        return len(self.data)
c = C([4, 5, 6])
len(c)
```

Question 2: Advanced Python

- (a) The following code will not print out the integers `[0, 1, 4, 9, 16]` as we might expect. Why not? How can we fix it? Why is `map` designed to work this way? [5]

```
map(lambda x: x**2, range(0, 5))
```

- (b) What is the time complexity of the following code, with respect to the length `n` of the list `L`? How can we use a different data structure to reduce it? [5]

```
def dupe(L): # detect duplicates
    n = len(L)
    for i in range(n):
        for j in range(i+1, n):
            if L[i] == L[j]:
                return True
    return False
```

- (c) Explain in simple terms how the `dict` data structure achieves $O(1)$ lookup performance. Consider only a typical lookup, ignoring the issue of collisions. [5]
- (d) Without using any import, implement a *memoised* version of this function. [10]

```
def fib(n):
    if n <= 1:
        return 1
    else:
        return n * fib(n - 1)
```

Question 3: Data Science

- (a) Consider Figure 1, below. Given only the two end-point values x_0 and x_n and the value n , how can we create a Numpy array of the values $(x_0, x_1, \dots, x_{n-1})$? [5]

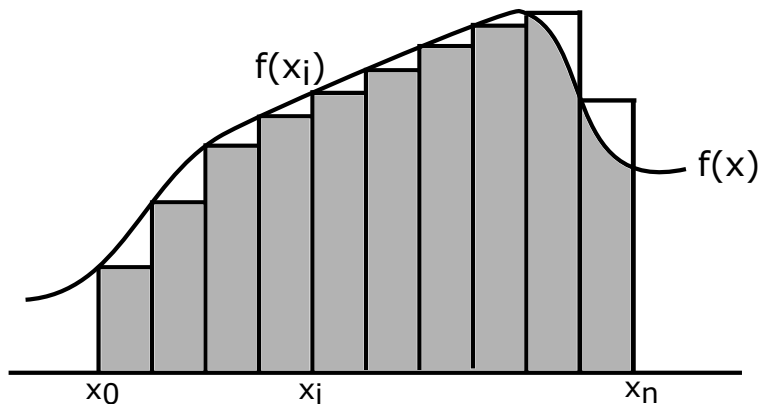


Figure 1

- (b) Referring again to Figure 1, above: suppose we are given a Python function f , and a Numpy array x of the values $(x_0, x_1, \dots, x_{n-1})$. We can approximate the area under the curve, between x_0 and x_n , by summing the area of rectangles as shown. The height of the i th rectangle is given by $f(x_i)$ as shown. Write a snippet of Python code or pseudocode, or explain in precise English how we can calculate this sum. [10]
- (c) What is a *mixin*? Explain with reference to an example in Scikit-Learn. [5]
- (d) For each of these R dplyr verbs, state its purpose in a few words: [5]
- `select`
 - `filter`
 - `mutate`
 - `gather` (or `pivot_longer`, if you prefer)
 - `inner_join`

Question 4: Tools and Applications

(a) Suppose we are given a *set* of variable assignments, such as:

- $x_1 = x_3 + 3$
- $x_0 = x_2 ** 2$
- $x_2 = x_3 + 1$
- $x_3 = 2$

Draw a graph representing the dependencies between the variables. State a *topological ordering* of the variables which would allow us to calculate their values. [5]

(b) For the following set, again draw a graph representing dependencies, and explain why no topological ordering is possible. [5]

- $x_4 = 2 * x_3$
- $x_2 = x_1 + 1$
- $x_1 = x_0 ** 2$
- $x_0 = 5 * x_2$
- $x_3 = x_2 + 1$

(c) State an algorithm for topological ordering, given a directed graph with nodes N and edges E . [5]

(d) Given the graph shown in Figure 2 below, write out the *adjacency matrix* which represents the graph. Use the adjacency matrix to calculate the out-degree of each node. [5]

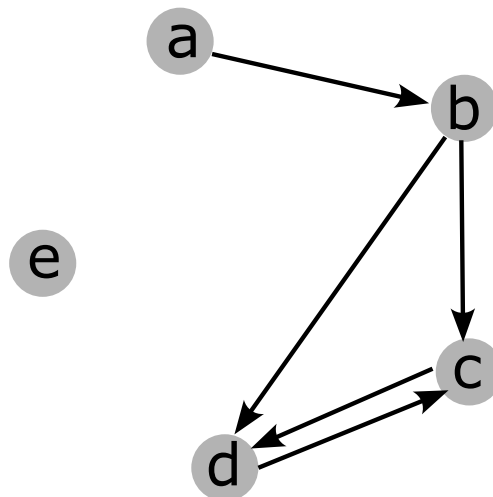


Figure 2

(e) In Assignment 2, we constructed a *dissimilarity matrix*. What properties does a dissimilarity matrix have? [5]