19,75 / 20 +0.5 → 20 / 20 ⭐ Congratulations!

IEFS Informatique et Société Numérique 2
SCAN, June 2024

INSA

Duration: 1h30
The use of any documents and calculators is prohibited
Please respond on the printout.
Note : A program with **poorly chosen variable names** will be **penalized.**

Modifier Pix :
-0,5 / 0 [✓ +0,5]

Last Name : BOCQUET First Name : Titouan Group : 62

Exercise 1 Network (2.5 pts)

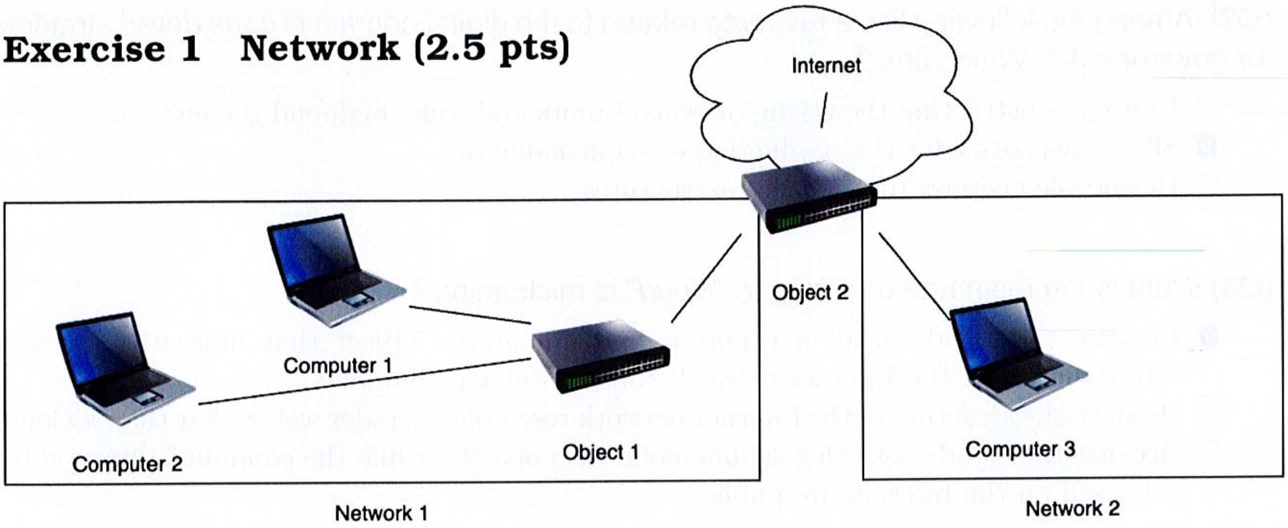


FIGURE 1 – Example of a network topology

E1 2,25

Consider the organisation of figure 1. There, *Object 1* interconnects *Computer 1*, *Computer 2*, and *Object 2* within the same network, while *Object 2* interconnects networks 1, 2, and the internet. Answer the following questions :

✓ **(Q1)** What type of device is *Object 1* ? 0.25

Object 1 is a ~~router~~ or a local switch.

✓ **(Q2)** What type of device is *Object 2* ? 0.5

Object 2 is also a router making a bridge between network 1 and 2.

✓ **(Q3)** What uniquely identifies a computer on a network ? 0.5

On a network the I.P. adress identifies a computer Internet-Protocol

✓ **(Q4)** Explain in a few sentences the role and functioning of the TCP protocol. 1,0

TCP is a communication protocol ensuring the good reception of all packets in the right order. Thanks to TCP, a server can request the client to resend data that has been lost along the way. ✓

E2
3

Exercise 2 Economic Model of the Web (3 pts)

2.1) Closed questions from the TD session *(1,5 pts)*

(0,5 pts if correct / -0,5 pts if incorrect / 0 pts if not answered. 1 answer / question)

(Q1) What is the *"two-sided market"* or *"bifacial market"* observable for online press ?

□ The online press sells content to its readers and pays its employees.

✓ ☒ The online press sells content to its readers and advertising *"spaces"* to advertisers.

□ The online press sells its content twice : at the time of publication and when consulting its archives.

(Q2) Among the following list, a resource related to the digital domain is considered strategic for governments. Which one ?

□ Iron for constructing the casing of central units and some high-end screens.

✓ ☒ Silicon necessary for the production of semiconductors.

□ Renewable energies to power infrastructures.

(Q3) What is the main reason the term *"cloud"* is misleading ?

✓ ☒ Because the cloud evokes a vaporous and immaterial object that hides the physical infrastructure of the Internet network and its ecological impacts.

□ Because the structure of the Internet network resembles a spider web rather than a cloud.

□ Because the cloud evokes fog synonymous with opacity, while the communication protocols used on the Internet are public.

2.2) Open Questions from the TD session *(1,5 pts)*

✓ **(Q4)** Name at least one problem that the concept of immateriality causes.

Privacy ; if we are not managing our own systems, somebody else need to do it : OVH fire a few years ago. We are not responsible for what they do on our machine.

✓ **(Q5)** How does digital technology disrupt the sovereignty of states ?

Big compagnies are monopolizing the digital world (GAFAM), most of them are hosted in the U.S. Therefore, a lot of our data travels through foreign lands.

✓ **(Q6)** Why do online services encourage Internet users to get more involved ?

They ask users to get more involve to collect information and then ~~propos~~ display more personalized advertisement to have a bigger income.

E3
2,0

Exercise 3 Course Question (2 pts)

✓

(Q1) Write a sorting algorithm of your choice (an algorithm, so no Python).
Recommendation : choose an iterative (non-recursive) algorithm and consider only sorting integers.

```
pivot = 0
while pivot did not reach the end of the list :
    i=0
    while number at index pivot + 1 greater that number at i and i < pivot
        add one to i
    save number at pivot + 1 as nb
    for j between (pivot +1 and i) going backwards
        replace numbert at j by number at j - 1
    replace number at i by nb.
    add one to pivot
```

not properly indented

✓

(Q2) What is the name of the algorithm you wrote ?

pivot sort

E4
8,5

Exercise 4 Recursion (8.5 pts)

4.1) Search in a List (2 pts)

2,0

(Q1) In Python, write a **recursive** function `is_present` that takes **at least** an unordered list `l` and an element `e` as parameters. It should return a boolean value `true` if `e` is present in `l` and `false` otherwise. (You do not need to write the docstring)

✓

```
def is_present(l,e,n)
    if n= len list :
        res = False
    elsif l[n] == e:
        res = True
    else
        res = is_present(l,e,n+1)
    return res

start at n=0.
```

if we were allowed to sort the list, could have done a dichotomy search more optimized

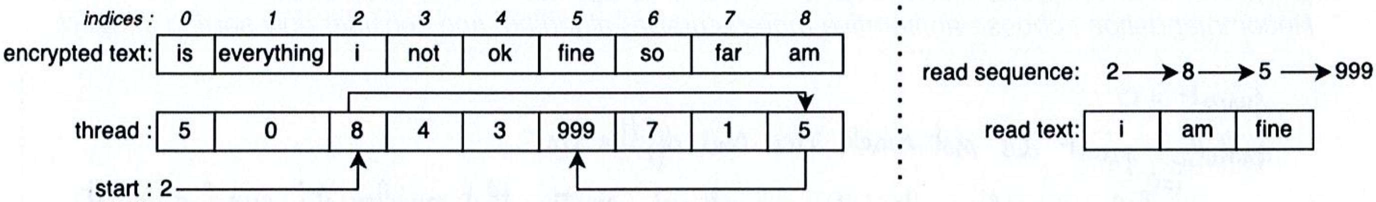
equivalent to a while loop:
while i<n
i+=1.

4.2) Text Decryption with 1 successor (4 pts)

Let's decode a text whose words are mixed up. To decrypt it, a "decryption thread" and a starting position are needed. Consider the example :

```
text = ["is", "everything", "i", "not", "ok", "fine", "so", "far", "am"]
thread  = [5, 0, 8, 4, 3, 999, 7, 1, 5]
start = 2
```

We decrypt it by reading the words in the order of the thread. The numerical value read from the thread is both the index of the next word and the next index. If this value is 999, it means there is no successor (end of traversal). This value is associated with an empty string :



indices :	0	1	2	3	4	5	6	7	8
encrypted text:	is	everything	i	not	ok	fine	so	far	am
thread :	5	0	8	4	3	999	7	1	5

start : 2

read sequence: 2 → 8 → 5 → 999

read text:		
i	am	fine

First the word at position 2 (start),
then the word at position 8 (value stored in 2 in the thread),
then the word at position 5 (value stored in 2 in the thread), ...

1,0 ✓

(Q2) Decrypt the text from the example using `start = 6` (1 pt)

so far everything is fine

(Q3) Write a recursive Python function `decryption` that returns the decrypted text. (3 pts)
(if you get stuck, write a non-recursive function (1 pt))

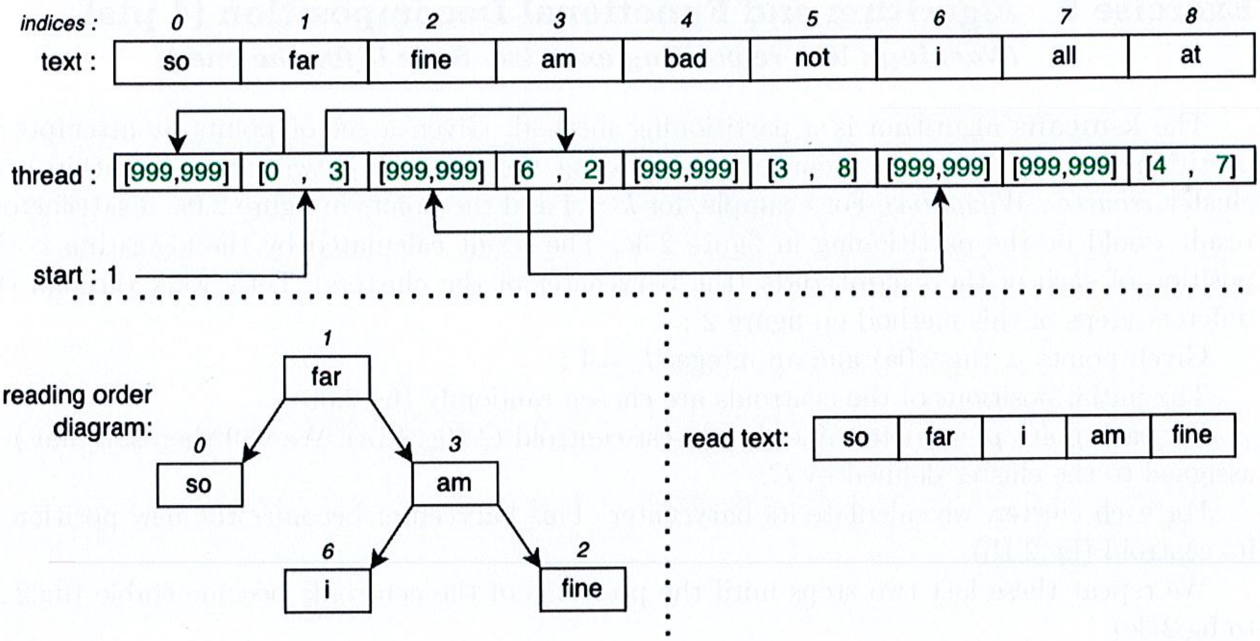
3,0 ✓

```
def decryption (txt, thread, start):
    if thread[start] == 999:
        res = txt[start]

    else :
        res = txt[start] + decryption (txt, thread, thread[start])
    return res
```

4.3) Text Decryption with 1 prefix and 1 suffix (2.5 pts)

Following the same principle as in the previous question, we now decide that the thread is no longer linear, but that each cell potentially has a prefix and a suffix. So now, the thread is a 2D list : each sublist contains exactly 2 elements, which are the text to be placed just before (cell 0) and the text to be placed just after (cell 1). If the value is 999, it means there is nothing before or after. Example :

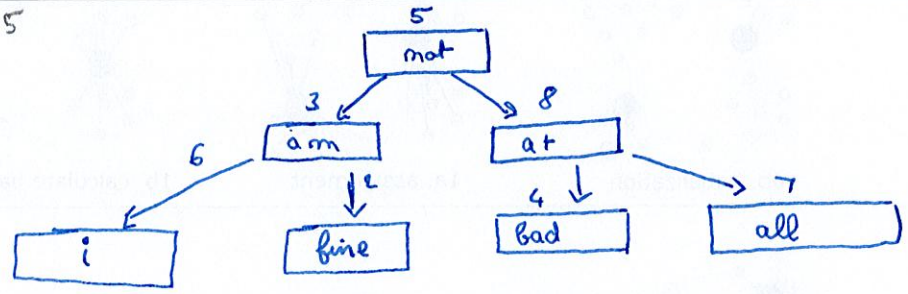


(Q4) Using `start=5`, draw the **diagram** of the reading order and **decrypt** the text from the example above. (1.5 pts)

1,5

✓

start = 5



I am fine not bad at all

(Q5) Write a recursive Python function `decryption_2d` that returns the decrypted text. (1 pt)

1,0

✓

```
def decryption_2d (text, thread, index)
    if thread[index] == [999,999]
        res = text [index]
    else :
        res = decryption_2d(text, thread, thread[index][0]) + text[index] + *
                  * decryption_2d (text, thread, thread[index][1])
    return res
```

assuming that end of the sequence from the left and from the right as in the example, no [-1,999] nor [999,8]

assuming space between words are already placed in the list "text"

E5
4,0

Exercise 5 Algorithm and Functional Decomposition (4 pts)

(Warning : less rewarding exercise. Save it for the end.)

The **k-means** algorithm is a partitioning method. Given a set of points, it attempts to partition the n points into k clusters, minimizing the distance between points within each cluster *(source : Wikipedia)*. For example, for $k = 4$ and the points in figure 2.0a, a satisfactory result would be the partitioning in figure 2.3c. The result calculated by the algorithm is the position of each of the k **centroids** (the barycenter of the clusters). Let's walk through the different steps of this method on figure 2 :

Given points p_i (fig.2.0a) and an integer $k = 4$:

The initial positions of the centroids are chosen randomly (fig.2.0b).

For each point p, we determine the closest centroid C (fig.2.1a). We will then say that p is assigned to the cluster defined by C.

For each cluster, we calculate its barycenter. This barycenter becomes the new position of its centroid (fig.2.1b).

We repeat these last two steps until the positions of the centroids become stable (fig.2.2a to fig.2.3c).

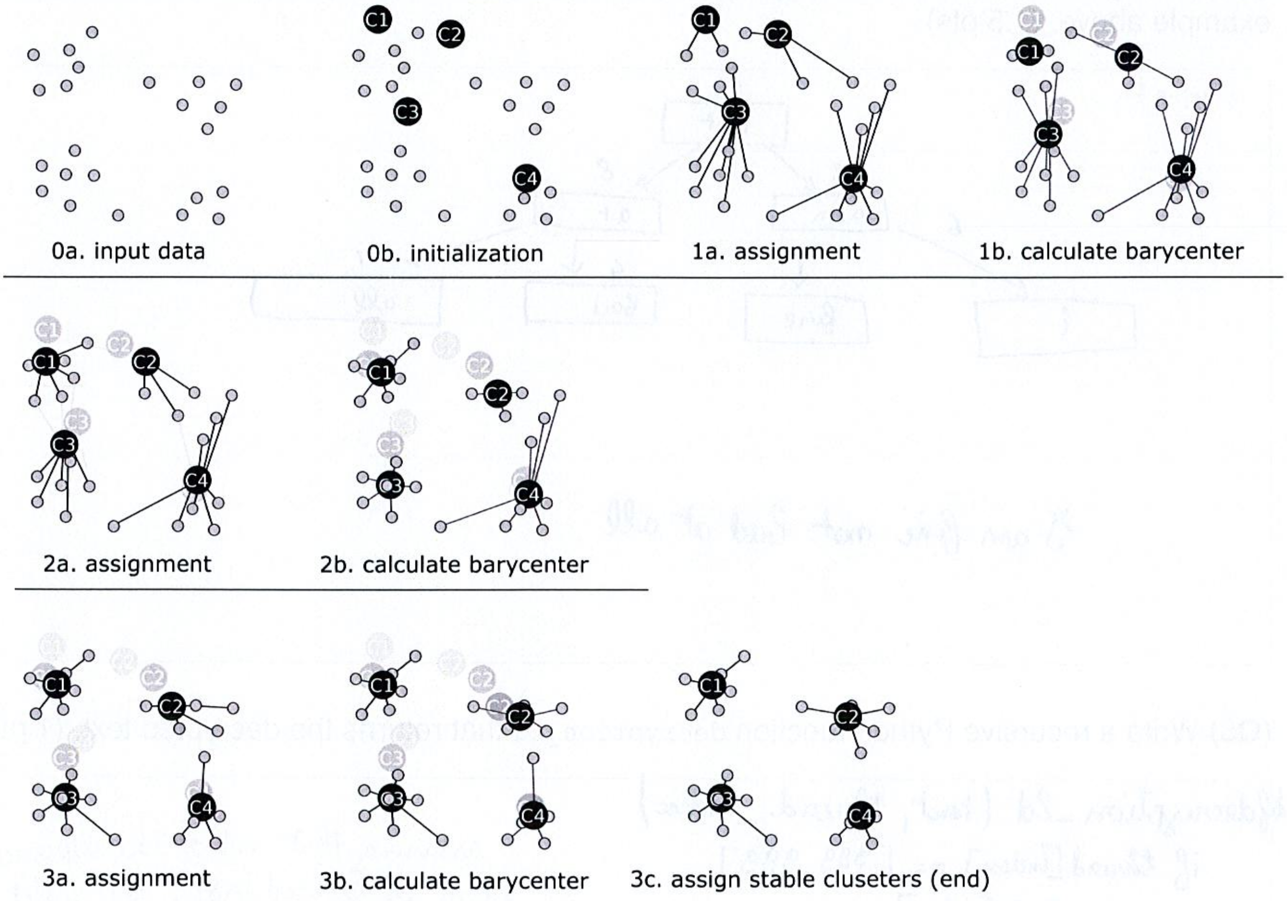


FIGURE 2 – Illustration du déroulement de l'algorithme des k-means en 4 itérations.
By Mquantin, CC BY-SA 4.0

1,0 ✓

(Q1) What are the inputs and outputs of this algorithm (parameters and result)? For each, propose an associated Python type (e.g., integer, string, 1D list of floats...) (1 pt)

Inputs : k number of clusters , a 2D list for the coordinates of the points (more if more dimensio)

Outputs : a 2D list for the positions of centroids , a 2D list to determine wich centroid is linked with which points.

(Q2) Without writing Python and without delving into fine details, write an **algorithm** for k-means clustering. Be sure to use the vocabulary given in the statement (point, centroid, partition, barycenter...). (2 pts)

2,0 ✓

(Volume indication : a good response written in natural language would be about ten lines)

randomly place k ~~barycenters~~ centroid.

link points to the closest centroid

compute the position of the bary center of each cluster .

place centroid at the computed bary center.

unlink centroid

repeat the last step until the position of the bary centers are only varying by a margin small enough.

1,0

(Q3) Perform a functional decomposition of your algorithm by identifying two Python functions that would be useful for implementing your algorithm in Python. For each, provide their **signature** and a **complete docstring** (the code for the function is not required) (1 pt)

```
def barycenter ( coord )
    ''' input :  coord: 2D list of the positions of the points
                        [[x1, y1], [x2, y2] ... , [xn, yn]] with n points.
    output :  a list of two numbers determining the coordinate
              of the bary center, defined as [vertical coord, horizontalcoord].
✓   function: A function that finds a barycenter of points that
              have the same weight by averaging '''

    return [x, y]
def randomplacement ( k, size )
    ''' input :  k int, the number of random points that will
                 be placed
             size :  the limit of the coordinate where the points will
✓                    be placed, first vertically, then horizontally
    output :  2D list of the random coordinates

    function :  randomly place k points in the limits delimited
                by size, returnthe list of coordinates of lenth k. '''
    return [[x1, y1], [x2, y2], ..., [xk, yk]].
```