Write your full name and your student ID/date of birth on top of all the pages

Mock Exam
Introduction to Data Science and Programming

Month Day, Year

Question	Points
1	10
2	10
3	10
4	10
5	10
6	10
7	10
8	10
9	10
10	10
Total:	100

Question 1: Data Structures (10 points)

Consider the following list of different lamps from an inventory, listing for each type of lamp the name and stock (how many are available). Suppose this list data structure contains n entries.

```python
stock_lamps = [
  ("henningsen", 87),
  ("panton", 53),
  ("poulsen", 8),
  ..., # the list continues here
  ("leklint", 35)]
```

(a) (5 points) Say we need to change the stock for one of the lamps. This lamp is identified by the string, say, `lamp = "nuura"` - we do not know its position in the list. In the worst case scenario, explain how long it would take, in relation to n, to perform this operation with this data structure? What about the best case scenario?

Solution: Worst case scenario is that the lamp we are looking for is the last tuple in both of the lists. In that case we have to look through the whole list, that is n operations, i.e., $O(n)$. The best case is that it's the first in both lists, then we get 2 operations which is $O(1)$.

(b) (5 points) Which Python data structure would be most efficient out of the data structures you learned to carry out the same task (without importing any library)? Explain, i.e. give the best and worst case amount of time in relation to n.

Solution: A dictionary is most efficient since it can access the count for a specific word in constant time if we set the key to be the lamp string and the value to be the count. Both best and worst case are $O(1)$ (if we exclude rare "amortized worst cases").

Question 2: Variables and Program Flow (10 points)

(a) Consider the block of Python code below.

```
i = 100
mylist = []
while True:
    modulo121 = i % 121
    if modulo121 == 0:
        mylist.append(i)
    i += 1
    if i > 3000:
        break
```

i. (2 points) Describe in your own words what this while-loop does, and what `mylist` will contain after you run the code. You don't need to give the exact numbers; a short description is sufficient.

Solution: It contains all integers between 100 and 3000 that are divisible by 121.

ii. (3 points) Write an implementation for the same functionality, i.e. producing a list `mylist` with the same content, with a for loop instead of a while loop.

Solution:

```
mylist = []
for i in range(100,3000):
    modulo121 = i % 121
    if modulo121 == 0:
        mylist.append(i)
```

(b) (2 points) In the code above, after the while-loop finishes, the variable `mylist` contains 24 items. Now instead, consider the block of Python code below.

```
i = 1
longlist = []
while True:
    modulo5 = i % 5
    if modulo5 == 0:
        longlist.append(i)
    i += 1
```

Can you estimate the length of `longlist`? Give a very short explanation.

Solution: No, it will fill up indefinitely as the while loop will never stop.

(c) Consider the block of Python code below.

```
s1 = True
s2 = False
s3 = True or False
s4 = True and False
all_statements = [s1, s2, s3, s4]
sum_statements = sum(all_statements)
long_statement = (s1 and s3) and (s3 or s2) and (s2 or s4)
```

i. (2 points) What is the variable type and the value of `sum_statements`?

Solution: int, 2

ii. (1 point) What is the boolean value of `long_statement`?

Solution: False

Question 3: Function Analysis (10 points)

Consider the block of Python code below.

```python
def isletter(character):
    letters = ["a","b","c","d","e","f","g","h","i","j","k","l","m","n","o","p","q","r","s","t"
                                                          ,"u","v","w","x","y","z"]
    if character in letters:
        return True
    return False

def myfunction(s):
    wordlist = []
    word = ""
    for i in range(len(s)):
        character = s[i]
        mode = len(wordlist) % 2   # modulo. The first mode will be 0.

        if isletter(character):
            if mode == 0:
                word = word + character
            elif mode == 1:
                word = word + "?"
            else:
                word = word + "%"
        elif character == ",":
            wordlist.append(word) # adding word to wordlist
            word = "" # starting next word

    if word != "":  # don't forget the last word
        wordlist.append(word)

    return wordlist

print(myfunction("*h8/ap_9py,ate*,n3e*_w,oh,y9ea/r"))
```

(a) (3 points) What does the code print out?

Solution: `['happy', '???', 'new', '??', 'year']`

(b) (5 points) Describe in your own words what the myfunction() function does to a string input?

Solution: It splits it into words separated by commas, stored in a list. Non-letter characters are ignored. Letters in every second word are replaced by question marks.

(c) (2 points) Which two consecutive lines of code inside the function myfunction() can be deleted without affecting the function's output?

Solution:

```python
            else:
                word = word + "%"
```

Full name: ______________________ **Student ID/Date of birth:** ______________________

Question 4: Induction (10 points)

Consider the equation below.

$$\sum_{i=1}^{n} i^3 = \frac{1}{4}n^2(n+1)^2$$

Prove, using induction, that the equation holds for all $n \geq 1$. Explicitly state the base case and the inductive assumption.

Solution: The equality holds for the base case $n = 1$:

$$\begin{aligned}\sum_{i=1}^{1} i^3 &= \frac{1}{4}1^2(1+1)^2 \\ 1 &= \frac{1}{4}4 = 1\end{aligned}$$

Let us assume it holds for $n = k$ (inductive assumption):

$$\sum_{i=1}^{k} i^3 = \frac{1}{4}k^2(k+1)^2$$

We check that then it holds for $n = k+1$:

$$\sum_{i=1}^{k+1} i^3 = \frac{1}{4}(k+1)^2(k+2)^2$$

It is true that $\sum_{i=1}^{k+1} i^3 = \sum_{i=1}^{k} i^3 + (k+1)^3$. Using the inductive assumption in the left hand side, we show the following equality:

$$\begin{aligned}\frac{1}{4}k^2(k+1)^2 + (k+1)^3 &= \frac{1}{4}(k+1)^2(k+2)^2 \qquad | : (k+1)^2 \\ \frac{1}{4}k^2 + (k+1) &= \frac{1}{4}(k+2)^2 \qquad | \times 4 \\ k^2 + 4(k+1) &= (k+2)^2 \\ k^2 + 4k + 4 &= k^2 + 4k + 4 \quad \square\end{aligned}$$

Full name: ______________________________ **Student ID/Date of birth:** ______________________________

Question 5: Data Exploration (10 points)

Consider the following data set with attributes and measurements about bicycle paths in Copenhagen.

pathID	width	flow	priority	protected
104	180	2400	1	1
105	120	3600	1	0
106	158	8800	1	1
201	178	10000	3	1
202	162	4600	2	1
203	220	8000	2	1
204	220	7400	1	0
205	240	7800	1	1
301	186	4600	2	1
302	140	12000	3	1
303	180	4600	2	1
304	182	9800	3	1
305	168	8800	1	1

(a) (2 points) Which of the variables are quantitative?

Solution: width, flow

(b) (3 points) Suppose you load the data into Python like this:

```
data = numpy.loadtxt("data.csv", skiprows=1, dtype="int")
```

The company *Hypercykelstier* is conducting a study on which of the bicycle paths could qualify as a “hyper bicycle path”. A path qualifies as “hyper” if it is protected and if at least one of these two conditions hold: Its width must be at least 160 cm, or it must have priority 1. Create a mask that selects all paths that qualify as “hyper”:

```
mask = (                    ) & ((                    )|(                    ))
```

where `data[mask,0]` should return the qualified pathIDs.

Solution:

```
mask = (data[:, 4] == 1) & ((data[:, 1] >= 160) | (data[:, 3] == 1))
```

(c) (5 points) Calculate the five number summary for the flows over all paths that qualify as “hyper” and report their sorted flows. If a five number summary value falls between two data points, take the midpoint (average) between the two data points.

Solution: Sorted weight values: 2400, 4600, 4600, 4600, 7800, 8000, 8800, 8800, 9800, 10000
Five number summary: min: 2400, Q_1: 4600, median: 7900, Q_3: 8800, max: 10000

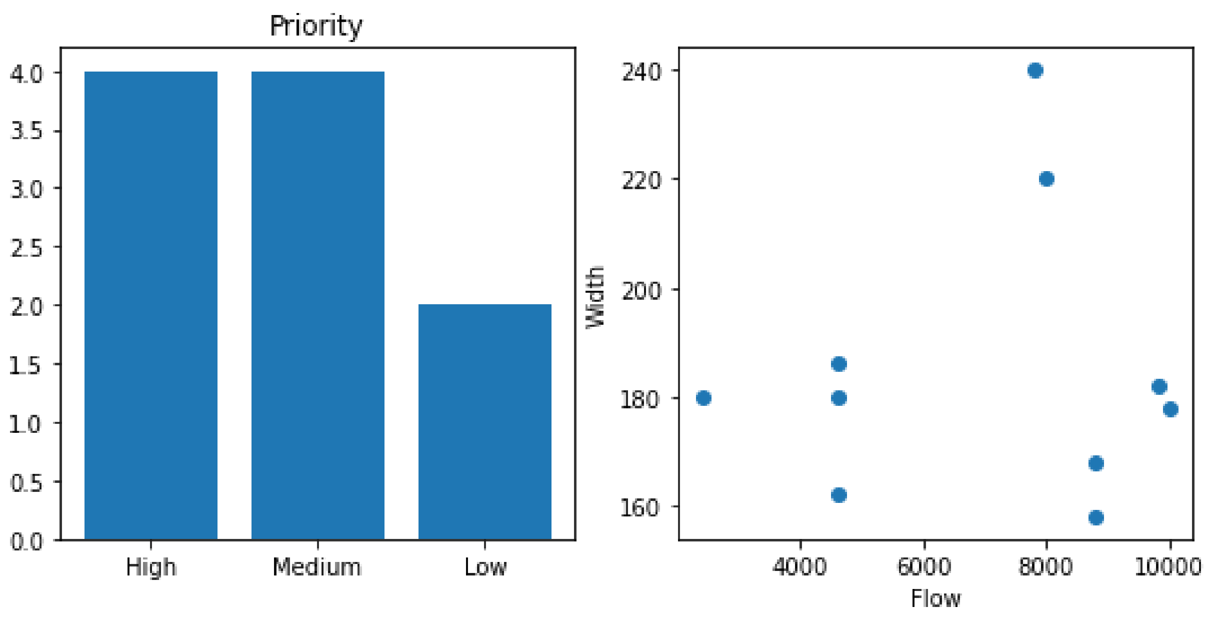
Full name: ______________________ **Student ID/Date of birth:** ______________________

Question 6: Visual Data Reporting (10 points)

Consider the data set `data_hyper = data[mask]`, where `data` and `mask` are taken from the previous question.

Draw by hand as exactly as possible what the following code will output. You can use the two boxes below the code for `axis1` and `axis2`. Draw all elements, including ticks, labels, titles, bars, scatterplot markers, at their correct locations. You do not need to get the colors right.

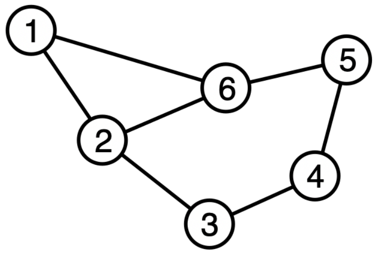
```python
import numpy as np
import matplotlib.pyplot as plt

categories, counts = np.unique(data_hyper[:, 3], return_counts=True)

fig, axes = plt.subplots(nrows=1, ncols=2, figsize=(9, 4))
axis1 = axes[0]
axis2 = axes[1]

axis1.bar(categories, counts)
axis1.set_title("Priority")
axis1.set_xticks(categories)
axis1.set_xticklabels(["High", "Medium", "Low"])

axis2.scatter(data_hyper[:,2], data_hyper[:,1])
axis2.set_xlabel("Flow")
axis2.set_ylabel("Width");
```



Full name: ______________________ **Student ID/Date of birth:** ______________________

Question 7: Network Metrics (10 points)

Given the following network.



(a) (2 points) Calculate the degrees $k_1, k_2, \ldots, k_6$, and the average degree $\langle k \rangle$ of the network.

Solution: $k_1 = 2, k_2 = 3, k_3 = 2, k_4 = 2, k_5 = 2, k_6 = 3, \langle k \rangle = \frac{2+3+2+2+2+3}{6} = \frac{14}{6}$

(b) (3 points) Calculate the clustering coefficients c_1, c_2 and c_3 of nodes 1, 2 and 3.

Solution: $c_1 = 1, c_2 = \frac{1}{3}, c_3 = 0$

(c) (4 points) Calculate the average path length ℓ and the diameter D.

Solution: $\ell = \frac{(1+2+3+2+1)+(1+1+2+2+1)+(2+1+1+2+2)+(3+2+1+1+2)+(2+2+2+1+1)+(1+1+2+2+1)}{30} = \frac{9+7+8+9+8+7}{30} = \frac{48}{30}$ = 1.6, $D = 3$

(d) (1 point) If there is a cycle of length 5 then report one below, otherwise write "No".

Solution: 2,3,4,5,6,2

Full name: ______________________ **Student ID/Date of birth:** ______________________

Question 8: Network Data Structure (10 points)

For the network from the previous question:

(a) (1 point) Write down the adjacency matrix.

Solution: $A = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 & 0 \end{pmatrix}$

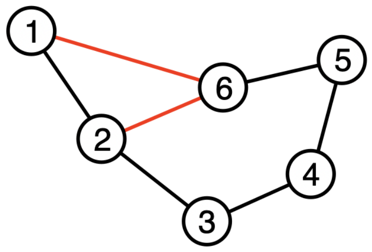
(b) (1 point) Write down the edge list.

Solution: $I = \begin{pmatrix} 1 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{pmatrix}, J = \begin{pmatrix} 2 \\ 6 \\ 3 \\ 4 \\ 5 \\ 6 \end{pmatrix}$

(c) (1 point) Write down the adjacency list.

Solution: $1 = \{2,6\}, 2 = \{1,3\}, 3 = \{2,4\}, 4 = \{3,5\}, 5 = \{4,6\}, 6 = \{1,2,5\}$

(d) i. (4 points) Remove from the same network below two links to increase the graph distance $d_{1,6}$ between the nodes 1 and 6 from $d_{1,6} = 1$ to $d_{1,6} = 5$. Indicate in the figure below where you remove the links.



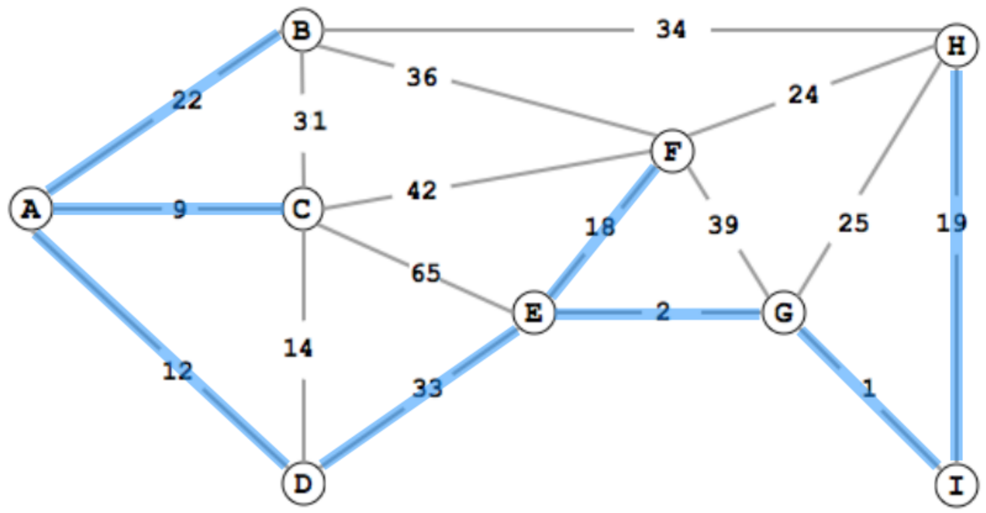
ii. (3 points) For this new network: What is then the new average degree, and what is the new diameter?

Solution: $\langle k \rangle_{\text{new}} = \frac{1+2+2+2+2+1}{6} = \frac{10}{6} = 1.67$, $D = 5$

Full name: ________________________ **Student ID/Date of birth:** ________________________

Question 9: Graph Algorithms (10 points)

Consider the following undirected weighted graph.

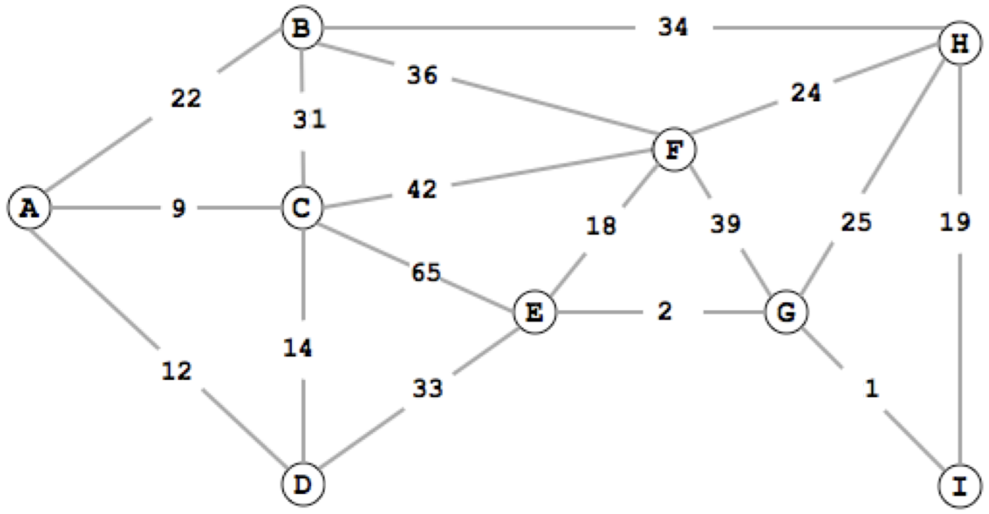


(a) (5 points) Find the minimum spanning tree of the graph by using Kruskal's algorithm. Highlight the minimum spanning tree edges in the figure above, and list the edges in the order of selection by the algorithm.

Solution: $GI, EG, AC, AD, EF, HI, AB, DE$

(b) (5 points) Find the minimum spanning tree of the graph by using Prim's algorithm, starting at node A. The same graph is plotted again for your convenience below. List the edges in the order of selection by the algorithm.

Solution: $AC, AD, AB, DE, EG, GI, EF, HI$



Full name: ______________________________ **Student ID/Date of birth:** ____________________________

Question 10: Command Line Tools (10 points)

Consider working with a UNIX terminal (i.e., the default terminal on a mac or in WSL).
In the following, lines starting with `%` are commands (or pipelines), other lines are outputs.

(a) Describe with one or two short sentences each command and its output.

i. (1 point)

```
% ssh misz@hpc.itu.dk
Welcome to hpc.itu.dk!
```

Solution: Connect to remote server. Welcome message from server.

ii. (1 point)

```
% cd data/
% ls
exam.csv      exam2.csv      raw
```

Solution: Switch to directory data, list folder contents.

iii. (1 point)

```
% wc -l exam.csv
5 exam.csv
```

Solution: Count the number of lines in file. Show number of lines and filename.

(b) Assume the following command is run with the output shown below it:

```
% cat exam.csv
id,grade,points,name
14353,10,86,smith
14233,12,99,vybornova
14533,10,84,pedersen
13633,02,58,szell
```

i. (2 points) What is the output that the following pipeline will generate?

```
% head -1 exam.csv | tr "," "\n" | sort | tr "\n" ","
```

Solution:

```
grade,id,name,points,
```

ii. (2 points) What is the output that the following pipeline will generate?
Note: `NR>1` means that the first line is skipped.

```
% awk -F',' 'NR>1 {print $2}' exam.csv | sort | uniq | tr "\n" " "
```

Solution:

```
02 10 12
```

iii. (3 points) Write a pipeline that creates a new file `names_sorted.txt` with the following content:

```
pedersen
smith
szell
vybornova
```

Solution:

```
% awk -F',' 'NR>1 {print $4}' exam.csv | sort > names_sorted.txt
```