BMIG 5003 Computational Methods for Biomedical Informatics
Horacio Gómez-Acevedo, PhD
Fall 2023
Final Test
Due day: Dec 11th, 12:00 pm CST

Submission:
All your answers should be placed in the Assigned Box Folder. I will accept only Python and text files, NOT jupyter notebooks. Feel free to use any following modules: `numpy, pandas, plotly, scikit-learn, statsmodels, pytest, bnlearn, and regex`.

1. Using `pytest`, generate a unit test for each of the functions in the program `FSM_simple.py` located in the teams folder. Submit your unit test file with the rest of your solutions.

2. A survey was conducted about the reliability of a new test for COVID-19. A precise medical testing classified the patients as healthy or sick and is encoded in the random variable *Status*. The area where the survey was conducted has 10% COVID positive prevalence during the survey time. The output of the test is encoded with the random variable X can have three outcomes: positive, negative, and inconclusive.

Status	X		
	Positive	Negative	Inconclusive
Healthy	0.08	0.80	0.12
Sick	0.82	0.09	0.09

Table 1: Distribution of the prediction of new test

   (a) Using Bayes' theorem. Calculate $P(\text{Status} = \text{Healthy}|X = \text{Positive})$.

   (b) Implement a Python program in which the user provides the prior prevalence distribution, and returns the posterior probability $P(\text{Status} = \text{Healthy}|X = x)$ for each value of x.

3. Consider the Bayesian belief network that classify fish (see Figure 1). The node A represents the time of the year and has four values: a_1 =Spring, a_2 =Summer, a_3 =Fall, and a_4 =Winter. The node B represents the location where the fish was caught and has values b_1 =North Atlantic, and b_2 = South Atlantic. The node X represents the type of fish and has values x_1 =Salmon, and x_2 =Sea bass. The node C represents the "lightness" of the fish and has values c_1 =light, c_2 =medium, and c_3 =dark. Finally, the node D represents the thickness and has values d_1 =wide, and d_2 =thin.

   (a) Calculate the join probability of the event that the fish was caught in the Spring, in the South Atlantic, and the fish was salmon and was light and thin.

   (b) Write a Python program that provides allows the user to give any event as in (a) and returns the joint probability for that event.

4. Suppose a terrain is represented by a two dimensional grid of elevation values. A peak is a grid point whose four neighboring cells (left, right, up and down) have strictly lower elevation values (see the corresponding graphical representation in figure 2). Write a Python program ask for the user for the size of the grid (say n by m), generates elevation randomly following a $N(0, 10)$ distribution. The output would be the number of

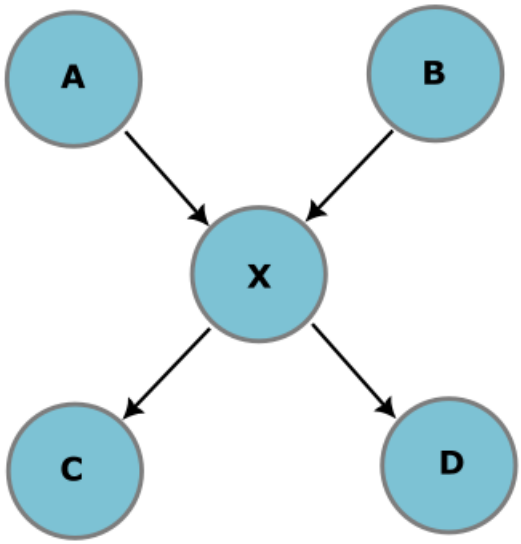

Figure 1: Bayesian Belief Network

$P(a)$	a_1	a_2	a_3	a_4
	0.25	0.25	0.25	0.25

$P(b)$	b_1	b_2
	0.6	0.4

$P(x\|a,b)$	$P(x_1\|a_i,b_j)$	$P(x_2\|a_i,b_j)$
a_1, b_1	0.5	0.5
a_2, b_1	0.7	0.3
a_3, b_1	0.6	0.4
a_4, b_1	0.8	0.2
a_1, b_2	0.4	0.6
a_2, b_2	0.1	0.9
a_3, b_2	0.2	0.8
a_4, b_2	0.3	0.7

$P(c\|x)$	$P(c_1\|x_i)$	$P(c_2\|x_i)$	$P(c_3\|x_i)$
x_1	0.6	0.2	0.2
x_2	0.2	0.3	0.5

$P(d\|x)$	$P(d_1\|x_i)$	$P(d_2\|x_i)$
x_1	0.3	0.7
x_2	0.6	0.4

Table 2: Conditional Probability Tables

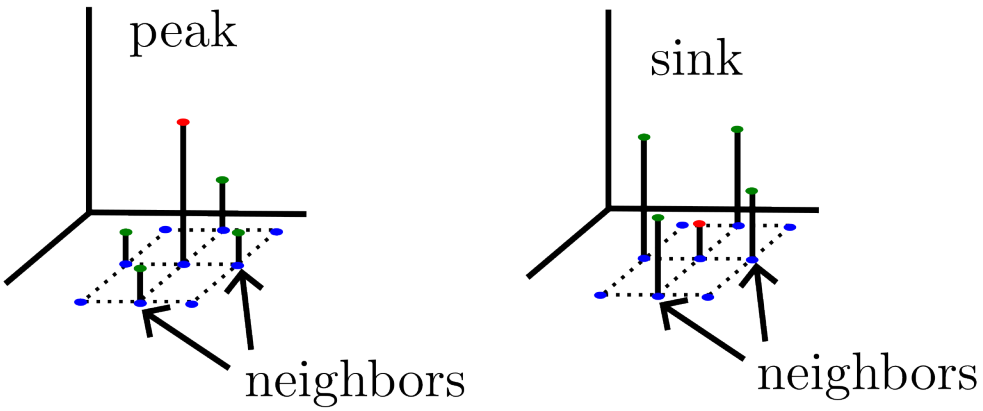

Figure 2: Peak and Sink points based on neighbors' values

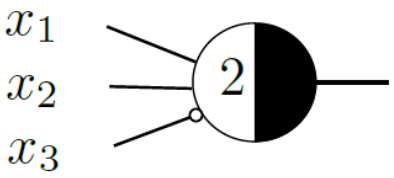

Figure 3: McCulloch Pitts unit

peaks and the location of each one of them. *Note. For the points located on the grid's border consider only the neighbors within the grid.*

5. Write a Python program that fills up the table where G is the function defined by the McCulloch-Pitts depicted in figure 3

x_1	x_2	x_3	$G(x_1, x_2, x_3)$
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

6. Given a sequence of elements $x_1, \ldots, x_n$ with a given (partial) order $\preceq$, we want to increasingly reorder the

sequence as $x_{i_1}, x_{i_2}, \ldots, x_{i_n}$, where $x_{i_k} \preceq x_{i_{k+1}}$ for $k \in \{1, \ldots, n-1\}$. For simplicity sake, you can think that we have a sequence consisting of numbers (say $\{17, 5, 8, 13, 3\}$) with the order given by $\leq$, thus the ordered sequence will be $\{3, 5, 8, 13, 17\}$ since $3 \leq 5$, $5 \leq 8$ and so on.

A simple sorting algorithm to reorder such a sequence follows the pseudo code

$a \leftarrow$ given sequence
while a is not sorted **do**
  for $i = 0$, until $i \leq length(a) - 1$ **do**
    if not $a[i] \leq a[i+1]$ **then**
      swap $a[i], a[i+1]$
    else
      continue
    end if
  end for
end while

You can define the sequence as a Python list `a=[17,5,8,13,3]`, and the steps are described in Figure 4. Write a Python program that accepts a sequence of numbers and orders them using your implementation of the previous pseudo code.

7. (Bonus) Expand your implementation in Problem 6 to order a sequence of points in the plane. More specifically, given a list of tuples $A = [(x_1, y_1), \ldots, (x_n, y_n)]$ with the order $(x_i, y_i) \preceq (x_j, y_j)$ if and only if $x_i \leq x_j$ and $y_i \leq y_j$. So, your program should return the ordered sequence. Note that not every sequence in the plane can be ordered, check your code with the sequence $[(9, 9), (8, 8), (7, 7), \ldots, (1, 1)]$. *Hint.* Do not overthink this problem, you just need to expand the tuple comparison and your *swap* function, the rest of the algorithm runs the same way.

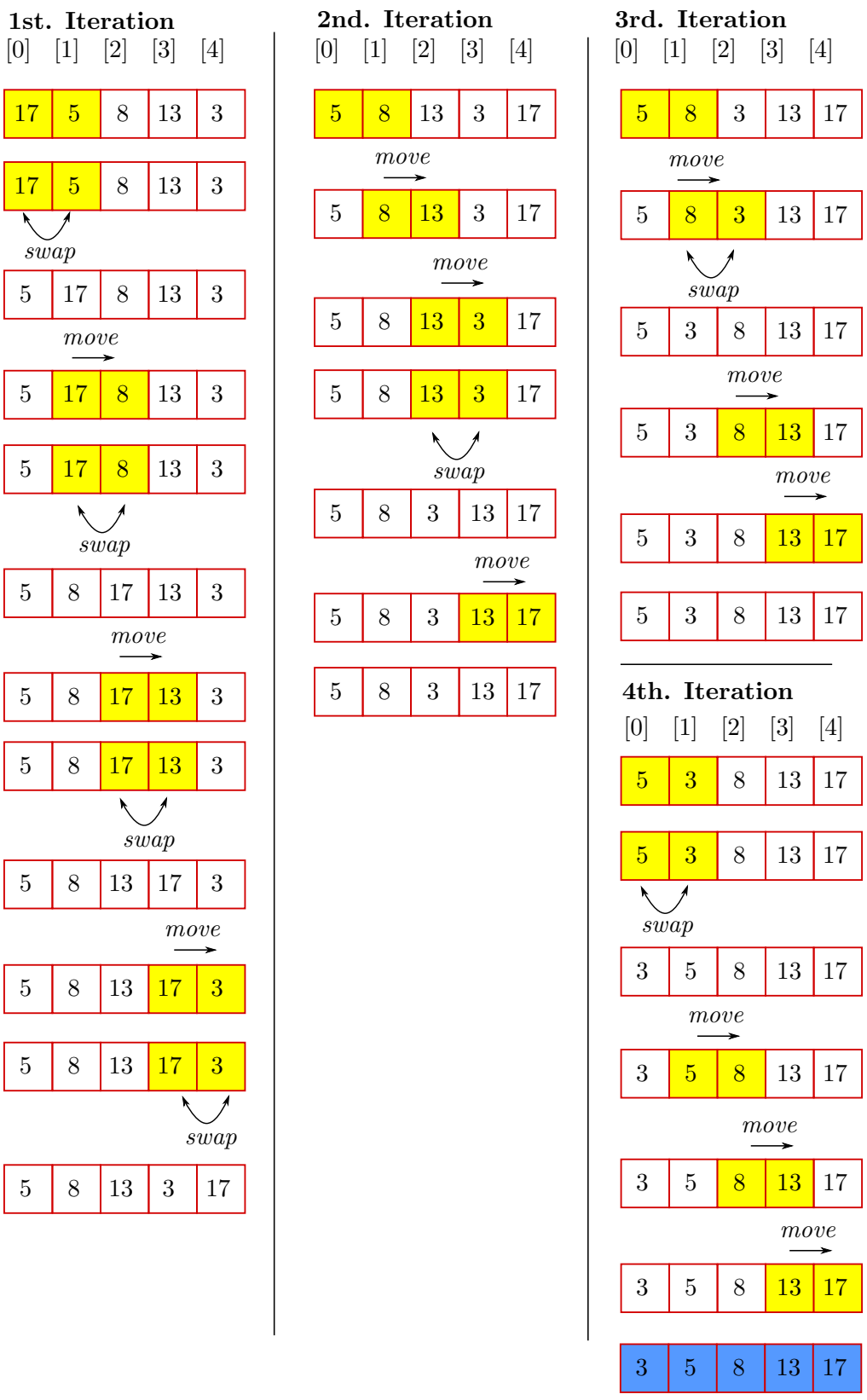

Figure 4: Sorting example