BMEn 2401 – Programming for Biomedical Engineers – Problem Set 8

Due, Monday, xx October 2024

Q1. Sorting is a process of rearranging a list of data in a new order. One common sorting algorithm is the bubble sort (see HV 10.3). The algorithm will repeatedly swap the adjacent elements if they are in the wrong order. Unlike the function y = bubble(x) in HV 10.3, the bubble sort can be implemented with- two **nested** for loops.

- Here is an example showing how the algorithm works.

The first iteration of the outer loop and iterations of the inner loop yields:
[5 2 1 3] -> [2 5 1 3] -> [2 1 5 3] -> [2 1 3 5] Now the last element is fixed

The second iteration of the outer loop and iterations of the inner loop yields:
[2 1 3 5] -> [1 2 3 5] -> [1 2 3 5] The last two elements are now fixed. Note that the last iteration of the inner loop does not swap anything since it is already in the right order.

The third iteration of the outer loop
[1 2 3 5] -> [1 2 3 5] Again no swapping is performed in this third iteration as it is already in the right order.

a. Write a function sorted_x = myBubbleSort(x) that takes an Nx1 row vector x and sorts it in ascending order using nested for loops as illustrated in the example above.
b. However, bubble sorting can be time-consuming because it requires iteration through the whole data set multiple times. There are other faster sorting methods such as the quick sort. This algorithm is applied in the MATLAB default sort(x) function and is more complicated than the bubble sort. The MATLAB built-in function B = sort(A) sorts the elements of A. By default, sort uses ascending sorted order.To compare the two sorting algorithms, write a function [bubble_time, merge_time] = compare(x). The function takes a LONGGG row vector x with random integers and outputs two variables bubble_time and quick_time, corresponding to the time needed to sort x for each algorithm. You will notice that the quick sort is much faster than the bubble sort. (Hint: use tic /toc handy functions in class to record the time used to run each algorithm)

Q2. In class, we learned that a complex number can be represented in polar coordinates, and that the complex number $\cos\theta + i\ \sin\theta$ represents a unit vector rotated an angle θ counterclockwise around the origin. In HW2 we had a problem using the rotation matrix to rotate a vector. In fact, a 2D rotation easily relates to a complex number:

For a random vector $\boldsymbol{x} = (x, y)$, we can write it in complex number form as $x + iy$. To calculate the rotated vector, we simply multiply the complex number by the unit vector $\cos\theta + i\ \sin\theta$.

For example, to rotate $\boldsymbol{x} = (1,1)$ (note: this is not a unit vector!) by $90°$ counterclockwise, we can convert it to $1 + i$ and then calculate the product $(1 + i) * (\cos 90° + i\ \sin 90°) = -1 + i$.The real portion will refer to the x coordinate and the imaginary portion will refer to the y. Therefore, the rotated vector $\boldsymbol{x_{rotated}} = (-1,1)$.

Based on the examples above, write a function x_rotated = rotation2D(x,theta) that takes a 1x2 row vector x and theta in degrees, and returns the rotated vector x_rotated. Round your answer to the nearest 100s place.

Q3. Patients and staff at your clinic have an awkward process when they come to be seen, and you would like to make it easier for the physician to pull the patient's information. The patient data is not collected in order, so you must put the data in the correct patient row.

a. In the function patientStruct = combinePatient(checkIn,weighIn,triage,ECG), create a structure named patientStruct with fields in this order: **Name Age Sex ECG Weight HR BP**

b. Using the input values checkIn, weighIn, triage, and ECG, fill in the corresponding sections of the structure.

-A patient checks in online with their name, age, and sex: checkIn.

-Age needs to be a number, not a string.

-A staff member takes the patient's weight: weighIn.

-Weight needs to be a number, not a string.

-A nurse will then come to take the patient's heart rate and blood pressure: triage.

-Heart rate needs to be a number and not a string.

-Blood pressure is given as two values (systolic and diastolic, respectively) but needs to be saved in the structure as a string with "/".

-If you get the values 120 and 80, your BP field should show "120/80".

-Some patients also have an ECG recording, but not all of them: ECG. The ECG array is in the order to match checkIn.

-To read the ECG array, use the following to convert it from a cell to a double:

-ecg = cell2mat(ECG(i));

c. Display the information for each patient.

d. If the patient has an ECG recording, plot that ECG. You can treat the empty values of ECG as a 0 when making your conditional statement.

e. The clinic decided that they want the weight to be converted from pounds to kilograms. Take the exsisting patient structure, patientStruct, and replace the weight measurements (originally in pounds) with the weight in kilograms. Do NOT make a new field, replace the existing one.

-Pounds to kg conversion: kg=lb/ 2.205