KSIAM-MathWorks Problem Challenge 2017

Problem 1 Which Doors are open?

There are n doors in an alley. Initially they are all shut. You have been tasked to go down the alley n times, and open/shut these doors in a particular manner:

*Toggle the state of each door you visit. That is, if the door is shut, open it, and if the door is open, shut it.

*On the i-th trip down the alley, start with door i and visit every ith door.

Example for case of n = 3:
On your first trip, you start from the first door and toggle every door. Now all three doors are open. On your second trip, you start from the second door and toggle every other door. Since there are only three doors, this means you shut the second door. Now doors 1 and 3 are open. On your third trip, you start from the third door and toggle every third door. Now only door 1 is open.

Write a MATLAB function that returns the list of open doors given n as an input.

Input: n
Output: y (array containing the indices of doors that are open, sorted as an increasing order)

Problem 2. Perfect Number

A perfect number occurs when the sum of all divisors of a positive integer, except the number itself, equals the number.

Examples:
28 is a perfect number because 28 = 1 + 2 + 4 + 7 + 14.
10 is not a perfect number because 10 ≠ 1 + 2 + 5.
Write a MATLAB function that determines the perfect number given n as an input.

Input: n
Output: y = true (if n is a perfect number) / false (if n is not a perfect number)

Problem 3. Kuramoto Model

The Kuramoto model, originally motivated by the behavior of chemical and biological oscillators, is a mathematical mode often used to describe synchronization of a networked system.

For given undirected graph G = (V, E), the state variable, originally representing phase of a limit-cycle oscillator, associated with the i-th node follows the dynamics:

$$\frac{d\theta}{dt} = \omega_i + \frac{K}{d}\sum_{j\in N(i)} (\theta_j - \theta_i), \quad i = 1,2,\ldots,M$$

where omega_i is a given intrinsic natural frequency, K is the coupling strength, d is the average degree of the network, and N(i) is the neighborhood of node i.
Write a Matlab function that constructs a d-regular graph from a given set of nodes such that each node is connected to their d closest nodes and then predicts the values of M state variables at t_final.

The function takes the form of:
Input: V, d, K, omegas, tspan, thetatInit
Output: thetaFinal
- V: M by n array whose i-th row indicates the coordinate of the i-th node
- d: degree of the network, which should be given less than M
- K: scalar to represent the coupling strength
- omegas: M by 1 array with parameters ω_i
- tspan: 1 by 2 row vector with [initial_time, final_time]
- thetaInit: M by 1 array with initial value of theta_i's
- thetaFinal: M by 1 array whose i-th entry represents the final value of theta_i

Note : 1. you can use MATLAB built-in ode45 for ODE solving. Do not use global variables.
2. example of a regular graph with degree 2 based on Euclidean distance node 1 (1,0); node 2 (0,1); node 3(-1,0); node 4(0,-1)
⇒ 1 connected to 2,4; 2 connected to 1, 3; 3 connected to 2,4; 4 connected to 1,3
⇒ every node is connected to the 2 closest nodes

Problem 4. K-means Clustering

The problem of K-means clustering is to group a given set of N data points in R^n into K groups such that each data point is closer to the group it belongs to than the centroid of any other group. Write a Matlab function that performs k-means clustering with N data points with a given initial guess and then predicts which cluster a new set of m test points belong to:

Input: XData, K, initCent, XTest
Output: idxData, centCluster, idxTest
- XData: N by n array of data points
- K: number of clusters
- initCent: K by n array of initial guess for the cluster centers
- XTest: m by n array of test points
- idxData: N by 1 array whose i-th entry is k if the i-th data point belongs to the k-th cluster
- centCluster: K by n array whose k-th entry represents the centroid of the k-th cluster
- idxTest: m by 1 array whose j-th entry is k if the j-th test point belongs to the k-th cluster

For your information, the most popular algorithm termed Lloyd's algorithm for this clustering takes an iterative of two steps: (a) assign data points to the best centroid amongst the K current centroids, (b) update the centroids with the assignments.

Note
1. You cannot use MATLAB built-in kmeans included in the Machine Learning toolbox.
2. If the resulting grouping is consistent, there is no restriction on the numbering of the clusters. In essence, centCluster (and the associated idxData) does not need to be sorted in a specific way.

Problem 5. The smallest integer with given number of divisors

For a given integer N (>0), P(N) is defined as the smallest natural number whose number of divisors equals to N. For example, the smallest positive integer with 16 divisors is 120, and P(16) = 120.

Calculate
$y = [\log_{10} P(2017)] + [\log_{10} P(2018)] + [\log_{10} P(2019)]$
$+ [\log_{10} P(2020)]$

Note that x is the largest integer that does not exceed x - you can use MATLAB function "floor".

input: x (Dummy Input)
output:
$y = [\log_{10} P(2017)] + [\log_{10} P(2018)] + [\log_{10} P(2019)]$
$+ [\log_{10} P(2020)]$

Problem 6. Signal Enhancement

Given the set of 16 signals, provide an enhanced signal (Noise Reduction)

Noise reduction are considered as an important problem in the modern electronic information communication age. Last 3 decades, there have been many researches to solve this problem, but still many issues not to be overcomed remains. In this problem, input signals are generated by adding 16 different noise. Therefore, you may try your own algorithm to single signal among 16 ones. Or you can consider a method using whole 16 signals. The evaluation is based on Signal to Noise Ratio (SNR)
https://en.wikipedia.org/wiki/Signal-to-noise_ratio.

Your mission is to surpass the given SNR threshold.

input: 2d array with size 16 by 64
output: 1d array with size 64