

Programming interview

Name: _____

Date: _____

Job applied for: _____

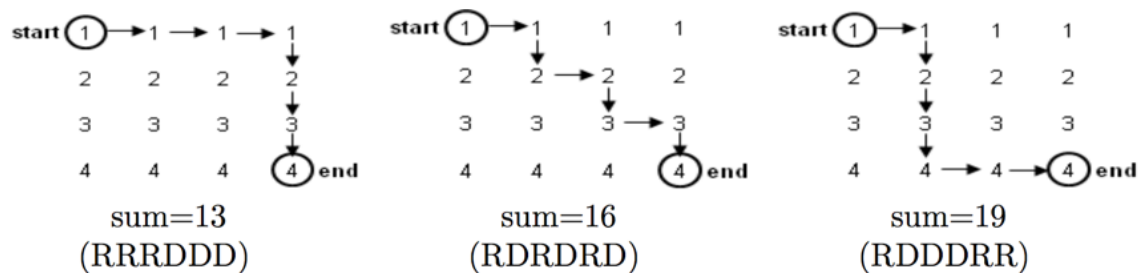
Tick the appropriate box: ☐ Citizen
☐ Permanent Resident
☐ Student Pass
☐ Others, please specify: _____

For each question, please submit your source code and output files in the required format according to the README and example output files. In addition, for questions requiring more explanation of your method, please write your explanations in the provided space in this question sheet. If you use any library functions, please also explain how the functions work.

1 Operations for the right sum

Given a $m \times n$ matrix, we want to connect from top left corner (starting point, first row, first column) to bottom right corner (ending point, m th row, n th column). Only 2 operations are allowed: Right (R) or Down (D). Numbers that are passed through will be summed up. Given any summed number, you are required to find out the operations needed to get the number.

Example: ($m=4$, $n=4$ square matrix) with operations needed to get the desired sum.



- For $m=9$, $n=9$ matrix, find the operations for the following summed numbers: 65, 72, 90, 110.
- For $m=90,000$, $n=100,000$ matrix (90,000 rows, 100,000 columns), find the operations for the following summed numbers: 87127231192 and 5994891682.

[Output file: output_question_1]

2 Equivalent networks

This question does not involve coding, please write your answers on a separate piece of paper.

2.1 Background on multi-layer perceptron

A multilayer perceptron (MLP) is a network which maps some input values to produce some output. The network consists of layers of nodes connected by weights. The input at the first layer is transformed layerwise to get the output. Figure 1 shows a single layer MLP with 2 input nodes connecting to 2 output nodes. The weights between the layers and the bias at each upper layer node controls the mapping. Assuming a linear activation function, the output values in figure 1 is given by

$$\begin{aligned} a_0^{(1)} &= w_{0,0}^{(1)}a_0^{(0)} + w_{0,1}^{(1)}a_1^{(0)} + b_0^{(1)} \\ a_1^{(1)} &= w_{1,0}^{(1)}a_0^{(0)} + w_{1,1}^{(1)}a_1^{(0)} + b_1^{(1)} \end{aligned} \quad (1)$$

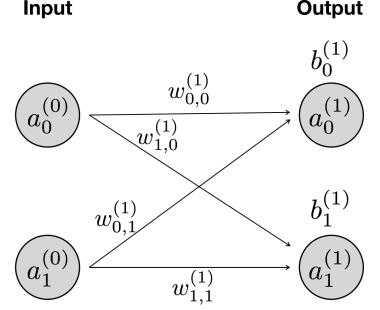


Figure 1: One layer MLP.

$a_i^{(l)}$: value of i^{th} node at layer l

$b_i^{(l)}$: bias acting on i^{th} node at layer l

$w_{i,j}^{(l)}$: weight connecting j^{th} node at layer $l-1$ to i^{th} node at layer l

Eq. 1 can be expressed more compactly using matrix and vectors, $\vec{a}^{(1)} = W^{(1)}\vec{a}^{(0)} + \vec{b}^{(1)}$, where $\vec{a}^{(0)} = (a_0^{(0)} a_1^{(0)})^T$, $\vec{a}^{(1)} = (a_0^{(1)} a_1^{(1)})^T$, $\vec{b}^{(1)} = (b_0^{(1)} b_1^{(1)})^T$, $W^{(1)} = \begin{pmatrix} w_{0,0}^{(1)} & w_{0,1}^{(1)} \\ w_{1,0}^{(1)} & w_{1,1}^{(1)} \end{pmatrix}$.

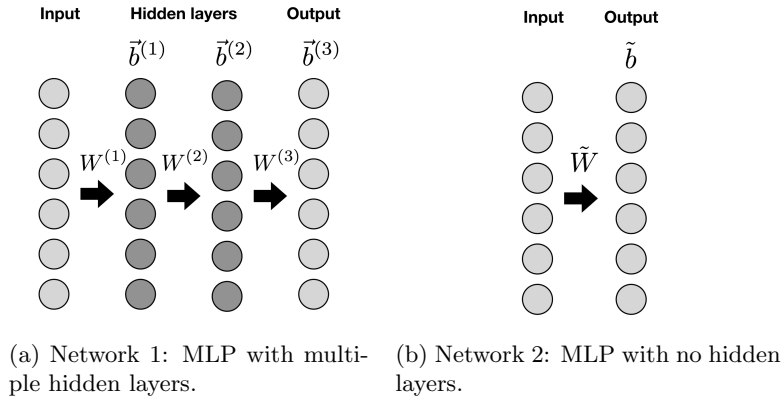


Figure 2

There can be any number of nodes in each layer. We can also have multiple hidden layers in between the input and output layers, where we use the same transformation equations. For the network in Figure 2a with two hidden layers, the output at the second and third layers are given by

$$\vec{a}^{(2)} = W^{(2)}\vec{a}^{(1)} + \vec{b}^{(2)} \quad (2)$$

$$\vec{a}^{(3)} = W^{(3)}\vec{a}^{(2)} + \vec{b}^{(3)} \quad (3)$$

2.2 Question: Formulating equivalent networks

Two networks are said to be equivalent if, they have the same number of input and output nodes, and for all inputs, the output of both networks are identical. Two neural networks are shown in Figure 2a and 2b with different number of hidden layers. Given Network 1's weights and biases values, find out Network 2's weights (\tilde{W}) and bias (\tilde{b}) such that the two networks are equivalent.

3 Multilayer perceptron for regression

Regression analysis is a method that is used for analysing the relationship between a set of independent variables and a dependent variable.

In this question, you are given a training dataset that consists of three independent variables x_1, x_2, x_3 and a dependent variable $y=f(x_1, x_2, x_3)$. You are required to construct the Multi Layer Perceptron (MLP) model shown in Figure 3 to predict the dependent variable y by using the values of independent variables x_1, x_2 and x_3 . After constructing and training the MLP model with training dataset, you will predict the values of y for the test dataset.

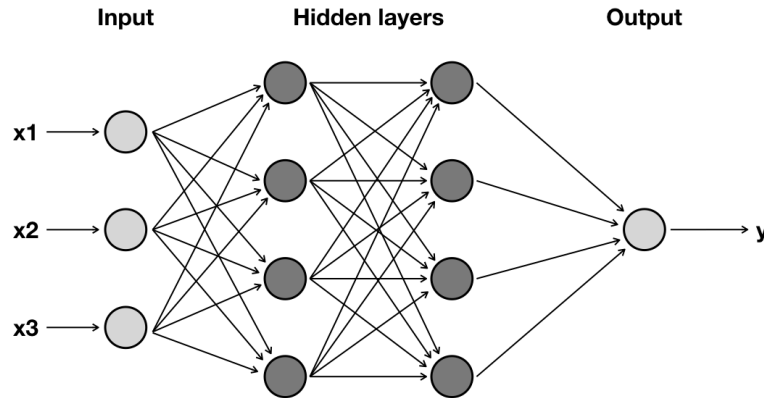


Figure 3: MLP Model: 3-input, 4x4 hidden layers and 1-output

[Input file1: train_data.txt (tab-seperated file: each column corresponds to an independent variable: x_1, x_2 and x_3 , respectively)]

[Input file2: train_truth.txt (ground-truth values of dependent variable: y)]

[Input file3: test_data.txt (tab-seperated file: each column corresponds to an independent variable: x_1, x_2 and x_3 , respectively)]

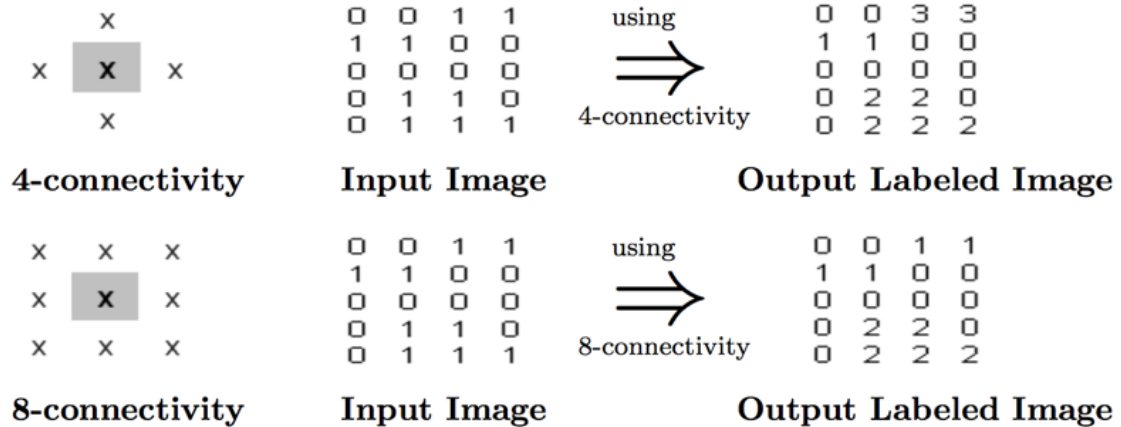
[Output file1: test_predicted.txt (write the predicted y values)]

Note: You can use Tensorflow, Keras and the other frameworks to construct your model.

4 Connected components

Write a code to find out connected components for a given image. When a group of pixels in the image is “connected” to each other, they are said to form a connected cluster and we refer to this cluster as connected component. In imaging, pixels can be connected in 4 neighbors (4-connectivity) or 8 neighbors (8-connectivity). In the example given below, the input image will result in 3 connected components if using 4-connectivity or 2 connected components if using 8-connectivity. You can implement either 4-connectivity or 8-connectivity connected components.

*Remember to explain the workings of your code. If you use library functions, please explain how they work.



[Input file: input_question_4; Output file: output_question_4]

5 Coloring

Given a L by L square grid, consider 4 neighbors connections. Given L^2 beads of different colors. Put all the beads onto the grid and the penalty for putting any two beads of the same color as neighbor is one. Total penalty is the sum of all penalties. Your task is to find a way to put the beads onto the square grid with least penalty. Perform this tasks with

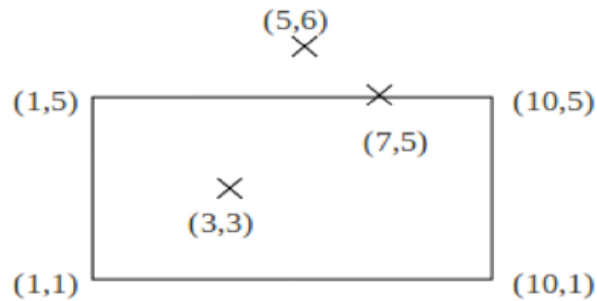
1. $L=5$ with 12 red beads (R) and 13 blue beads (B)
2. $L=64$ with 139 red beads (R), 1451 blue beads (B), 977 green beads (G), 1072 white beads (W), 457 yellow beads (Y)

[Output file1: output_question.5_1 (write the grid configuration (bead placement) for part 1)]

[Output file2: output_question.5_2 (write the grid configuration (bead placement) for part 2)]

6 Points inside/outside polygon

Given a sequence of points that form a polygon, you are required to tell if a list of points are either inside or outside the polygon.



Example: Given a sequence of points for polygon: (1,1),(1,5),(10,5),(10,1). The following is the outcome of points tested.

Point	State
(3,3)	Inside
(7,5)	Inside
(5,6)	Outside

*Remember to explain the workings of your code. If you use library functions, please explain how they work.
[Input file: input_question_6_polygon, input_question_6_points; Output file: output_question_6]

7 Coordinates-to-index & Index-to-coordinates

This question involves both deriving mathematical equations and coding, please write your derivations on a separate piece of paper. **For this question, do not use build-in library functions, code all the implementations.**

7.1 2-dimension

For one dimensional grid, indexing the cells is just going along one direction (1-dimensional raster scan) as shown in Figure 4a. For two dimensional grid, indexing the cells is done by 2-dimensional raster scan as shown in Figure 4b. For example, a 2-dimensional grid with sizes $(L_1, L_2)=(4, 3)$ is shown in Figure 4b. In this grid, coordinates $(x_1, x_2)=(2, 1)$ corresponds to index $I = 6$, and vice versa.

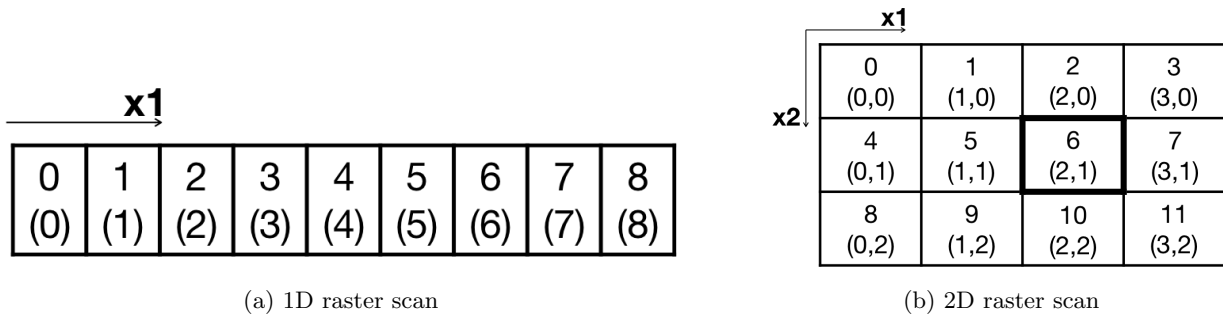


Figure 4

a) Derive a mathematical equation converting coordinates to index and derive the inverse equations converting index into coordinates in 2-dimensional grid.

b) Given 2-dimensional grid with sizes $(L_1, L_2) = (50, 57)$, write a code by yourself to do this (**do not use library functions, code all the implementations**):

- Write a code to convert given coordinates to index
(i.e. given x_1 and x_2 , find index I)

[Input file1: input_coordinates_7.1.txt (tab-separated file: each column corresponds to a dimension)]
[Output file1: output_index_7.1.txt (write the calculated index values)]

- Write a code to convert given index to coordinates
(i.e. given I , find coordinates x_1 and x_2)

[Input file2: input_index_7.1.txt]
[Output file2: output_coordinates_7.1.txt (write the calculated coordinates)]

7.2 d-dimension

For a grid of d-dimension with sizes $L_1, L_2, L_3, \dots, L_d$, indexing is done by d-dimensional raster scan.

For example, a 3-dimensional grid with sizes $(L_1, L_2, L_3)=(4, 3, 2)$ is shown in Figure 5. In this grid, coordinates $(x_1, x_2, x_3)=(2, 0, 1)$ corresponds to index $I = 14$, and vice versa.

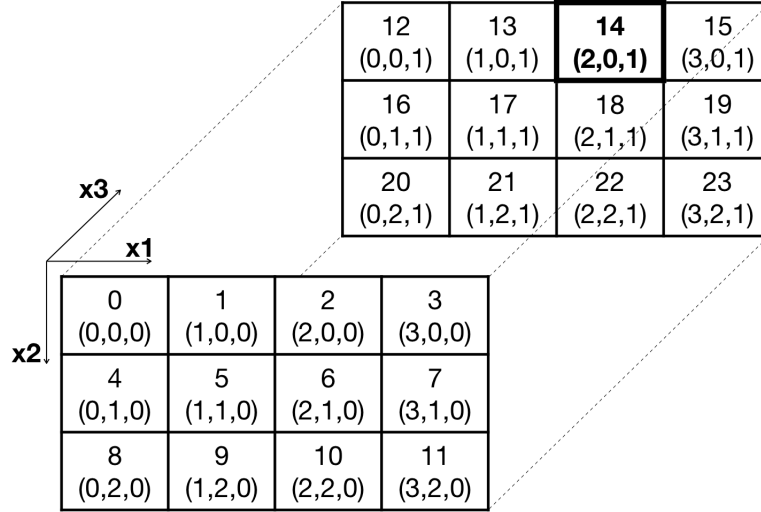


Figure 5: A 3-dimensional grid with sizes $(L_1, L_2, L_3)=(4, 3, 2)$.

a) Derive a mathematical equation converting coordinates to index and derive the inverse equations converting index into coordinates in d-dimensional grid.

b) Given 6-dimensional grid with sizes $(L_1, L_2, L_3, L_4, L_5, L_6)=(4, 8, 5, 9, 6, 7)$, write a code by yourself to do this (**do not use library functions, code all the implementations**):

- Write a code to convert given coordinates to index
(i.e. given $x_1, x_2, x_3, x_4, x_5, x_6$, find index I)

[Input file1: input_coordinates_7.2.txt (tab-separated file: each column corresponds to a dimension)]
[Output file1: output_index_7.2.txt (write the calculated index values)]

- Write a code to convert given index to coordinates
(i.e. given I , find coordinates $x_1, x_2, x_3, x_4, x_5, x_6$)

[Input file2: input_index_7.2.txt]
[Output file2: output_coordinates_7.2.txt (write the calculated coordinates)]