

AIR UNIVERSITY ISLAMABAD CAMPUS
OPERATING SYSTEMS (CS-325)
FALL 2024
ASSIGNMENT-2
CLO- 2 PLO-3

Due Date: December 01, 2024 (11:59 PM)

Marks: 100

Instructions:

1. *Make sure that you read and understand each and every instruction.*
2. *It is a group assignment of group size = 3. No more or less than 3 members are allowed in a group.*
3. *Plagiarism is strongly forbidden and will be very strongly punished. If we find that you have copied from someone else or someone else has copied from you (with or without your knowledge) both of you will be punished. You will be awarded straight zero in this assignment.*
4. *Your submission must be a single zip file. No rar or any other format is allowed. The file must be renamed with the roll numbers of your group members section. For example, if your group member's roll numbers are 22i-1234, 22i-1235, 22i-1236 and your section is B, so, the file must be named as 22i-1234_22i-1235_22i-1236_B. Only one submission per group is required and that must be done by the group representative of that group.*
5. *Be ready for a viva/demo from this assignment. During the demo, every group member must be able to explain each and every line of the code.*

Question 1: Receipt generation using Threads

You are required to do multithreading on generated receipt. You will create 4 threads.

Thread 1: select the number of items you purchased. Display the prices of all items as (Quantity x Price of the individual) add them and return their sum.

Thread 2: Calculate 8% tax on it.

Thread 3: If an order is greater than 250, Calculate 10% sale.

Thread 4: Sort the items according to their prices.

Output:

Thread 1:

Items Purchased = eggs, bread, and chocolate.

eggs = 3 x 15=45, bread = 1x 60 =60, chocolate = 5 x 50 = 250.

sum = 45+60+250= 355

Thread 2:

355/ (1+0.8) = 197.22, taxed sum=197.22+355= 552.33.

Thread 3:

price=355, sale =0.1, 355*0.1=35.5, 355-35.5 = 319.5

Thread 4:

Items	Price
Chocolates	250
Bread	60
eggs	45

Question 2: Forward Propagation in Neural Network

You will do Forward propagation of Neural Networks using Multiprocessing and multithreading.

Explanation of Forward Propagation of Neural Networks:

It consists of 3 steps:

Step 1: multiply the weight matrix W with the features matrix x_i .

Step 2: add the result of step 1 with biases matrix b . the result will be called Z .

Step 3: Apply the sigmoid function on all elements of the resultant matrix from step 2.

$$\text{Sigmoid function is } A = \frac{1}{(1+e^{-Z})}$$

So, create three processes P1, P2, and P3. P1 will create multiple threads to multiply the weight matrix and biases matrix. (Matrix Multiplication using multithreading). The number of Threads in P1 will be equal to the number of rows of weights W .

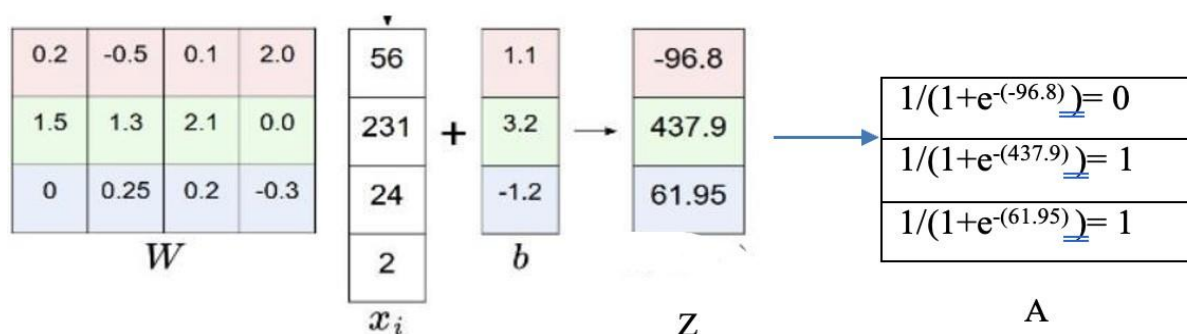
P2 waits for its completion and uses the resultant matrix to add its biases matrix using multithreading (Matrix Addition using Multithreading). The number of Threads in P2 will be equal to the number of rows of biases matrix b .

P3 waits for its completion and apply the sigmoid function on all elements of the resultant matrix from P2. P3 will also use multithreading to apply the sigmoid function. The number of Threads in P3 will be equal to the number of rows, resulting from P2. All threads work simultaneously to calculate the sigmoid function.

Hints:

- Use exp function present in cmath library to compute exponent. For example, exp(5) will calculate e^5 .
- Use pow function present in cmath library to compute the power. For example pow(4,5) will calculate 4^5 .

Sample Example:



Question 3: Creating an Image filter using Threads

You have to send an image to an LCD of Arduino, but your image is not clear. Before displaying image on LCD, you have to perform some operation to remove noise from it.

Image file is provided to you. It contains information about the pixels of an image. This image is a grayscale image, so pixel information contains level of different grayscales. Levels of grayscale are from 0 to 255. Dimension of this image is 16x16. Information of this image is available in "Matrix.txt" file. Copy and paste this array in your code. Now perform following operations on this image.

1. Print the frequency table of all grayscales. That is known as "Histogram" of an image.

- a) Count the frequency of each gray scale 0-255. E.g if grayscale"0" is appearing 10 times in the matrix of the image then it has frequency 10. Use 16 threads to calculate frequency of grayscales.

For Example:

Grayscale	Frequency
0	10
1	3
.	
.	
.	
255	45

2. Calculate the mean of the image matrix.
- a) Logically divide your matrix into 4X4 block. Compute mean value of each block. Like shown in the following table. Each block will be divided by a thread which means there will be 16 threads for this scenario.

4x4 Block 0	4x4 Block 1	4x4 Block 2	4x4 Block 3
4x4 Block 4	4x4 Block 5	4x4 Block 6	4x4 Block 7
4x4 Block 8	4x4 Block 9	4x4 Block 10	4x4 Block 11
4x4 Block 12	4x4 Block 13	4x4 Block 14	4x4 Block 15

- b) Print mean value of each block using threads.
- c) Print all block numbers which have a difference of less than 1 from the mean value of the original image using threads. Start block numbers from 0.
- d) Traverse all block again using threads. Replace pixel value in each block on the index that is equal to the block number with the mean value of the image.
- E.g., On block#0 replace Row#0, Col#0(Pixel#0)
 - E.g., on block#5, replace Row #1, Col#1(Pixel#5)
- e) Again, print the frequency table of the new modified image using threads.

Question 4: Travelling Salesman Problem

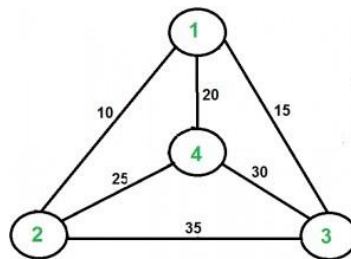
Travelling Salesman Problem (TSP): Given a set of cities and distance between every pair of cities, the problem is to find the shortest possible route that visits every city exactly once and returns to the starting point.

For example, consider the graph shown below. A TSP tour in the graph is 1-2-4-3-1. The cost of the tour is 10+25+30+15 which is 80.

The data is provided in file showing distance between all the nodes. The number of nodes can vary but make sure to attach the testFile while submitting your code.

testFile.txt

```
4
1 - 2 10
1 - 3 15
1 - 4 20
2 - 3 35
2 - 4 25
3 - 4 30
```



Take a number N from user and create N different threads. Each thread will find a solution of TSP by randomly generating numbers (Make sure the sequence of random numbers generated by different thread is not the same). Now each thread will save their output (cost of the tour) in a global array (after displaying it on terminal along-with their thread id). When all threads will be done computing the value, they will check whether their value was the minimum cost or not. The threads whose solution was not shortest will be stuck in an infinite loop. The one with shortest solution will output the thread id and cancel all the remaining threads.

If two threads have the same value, then define a mechanism as tiebreaker and explain it in comment on the top of your code file.

Note:

Submit your own work. Your efforts will be valued but copy pasting will be awarded a straight ZERO in the assignment.

Good Luck!!! ☐