

Due: April 9

Instructor: Professor Vijay K. Garg (email: garg@ece.utexas.edu)

The goal of this assignment is to learn LLP and CUDA programming. For each of the problems in this assignment use an efficient parallel algorithm.

1. **(50 points)** Apply LLP Algorithm for the following problems:
  - (a) Finding connected components of an undirected graph
  - (b) Bellman-Ford Algorithm
  - (c) Johnson's algorithm for shortest path
  
2. **(10 points)** Write a parallel program in cuda that reads a text file "inp.txt" and performs various computations on the data in the file. The output is in files: "q1a.txt" and "q1b.txt." The input file contains a list of integers in the range [0-999] separated by commas. Your program should read this file in an array  $A$  of integers. The output has the same syntax.
  - (a, 5 points) Compute  $\min A$ , the minimum value in the array.
  
  - (b, 5 points) Compute an array  $B$  such that  $B[i]$  is the last digit of  $A[i]$  for all  $i$ .
  
3. **(20 points)** Read an array  $A$  as in the first question. The output is in files: "q2a.txt," "q2b.txt," and "q2c.txt."
  - (a, 5 points) Create an array  $B$  of size 10 that keeps a count of the entries in each of the ranges: [0, 99], [100, 199], [200, 299], ... For this part of the problem, maintain array  $B$  in global memory of GPU.
  
  - (b, 5 points) Repeat part (a) but first use the shared memory in a block for updating the local copy of  $B$  in each block. Once every block is done, add all local copies to get the global copy of  $B$ .
  
  - (c, 10 points) Create an array of size 10 that uses  $B$  to compute  $C$  which keeps count of the entries in each of the ranges: [0,99], [0,199], [0,299], ..., [0, 999]. For this part of the problem, you must not use array  $A$ .
  
4. **(20 points)** Read an array  $A$  as in the first question. Compute an array  $D$  such that  $D$  consists only of odd numbers in  $A$ . You would need to determine the total number of odd numbers in  $A$ , and then copy all the odd numbers from  $A$  to  $D$  preserving their order in  $A$ . The output is in file: "q3.txt."