CS 573300 Assignment 2
Data Processing in **GPU**

1. Problem Formulation

In this assignment, you need to implement a program to solve the *k*-nearest neighbor (**KNN**) problem in CUDA. The KNN problem is described as follows. **Given *m* nodes in a *n*-dimension space, for each node, output a set *S* of another k nodes, such that no any other node neighbors**.

The distance is defined using Euclidean distance. For example, for node v at position (v_1, v_2, ..., v_n) and node u at position (u_1, u_2, ..., u_n), their distance is defined as

d(u,v) =[sum_{i=1,..n} (v_i-u_i)^2]^{1/2}

2. Data Sets

You have to proccess two data sets:

1. https://archive.ics.uci.edu/ml/datasets/Combined+Cycle+Power+Plant
2. https://www.kaggle.com/abcsds/pokemon

Please read the descriptions in the websites carefully. Note that all the **non-numerical attributes** in the data sets should **NOT** be take into account in your computation work.

3. Requirements

You should provide **a report** with your design ideas, implementations, and clear building steps for TAs to follow.

Your should **pre-process** each data set into a plain-text file such that all columns are space-separated.

Your program should accept two arguments, so that a data set can be processed as:
$ ./your_program <data file name> <the value of k>

The output should be a space-separated plain-text file, with **each line containing k sorted indexes**, which represent the data entries that are k-nearest to the corresponding entry. E.g. Given k=2, a data set with 5 entries may looks like this (comments are not really there):

```
1 3  // the 0th entry is closer to 1 and 3 than 2 and 4
0 3
0 4  // one of the entry 0 or 4 is the nearest neighbor to entry 2,
0 1                                    // and the other one is 2nd-nearest.
0 2
```

You should freeze all your source files after the dealine. The report should be uploaded to ILMS.

4. Evaluation

- (50%) Your program generates correct outputs.
  - (35%) CUDA code(s)
  - (15%) Report: your design and implementation in detail
- (20%) Your program exploit the parallelism of GPU, i.e using multiple blocks and threads.
- (15%) Do coalesced access to global memory as best as you can.
- (10%) Use shared memory to optimize the performance.
- (5%) Other notable optimizations. Please note these in your report

5. Account and Workstation Information

1. Prepare a SSH client software such as mobaXterm.
2. Connect to server 140.114.91.19
   1. Your login ID would be **your student ID**
   2. The password is **cs573300**
3. Setup a CUDA environment variables
   1. add /usr/local/cuda/bin into PATH
   2. add /user/local/cuda/lib64 into LD_LIBRARY_PATH
4. You should be able to create simple hw.cu, use nvcc to compile the CUDA code, and run now.