Assignment 3

Question 1:

This program is designed to find the minimum distance between two points among a set of points scattered in a 2D space. Using CUDA and a GPU we distribute calculating the minimum distance in a very large number of threads. Without using a GPU kernel if the program is executed on 1 process on a CPU it will take a very long time as it will exhaustively check all the combination of particle pairs and the time complexity will be $O(n^2)$. As GPU kernels parallelise this computation using a very large number of threads, the time complexity is effectively reduced.

The efficiency of approaches.

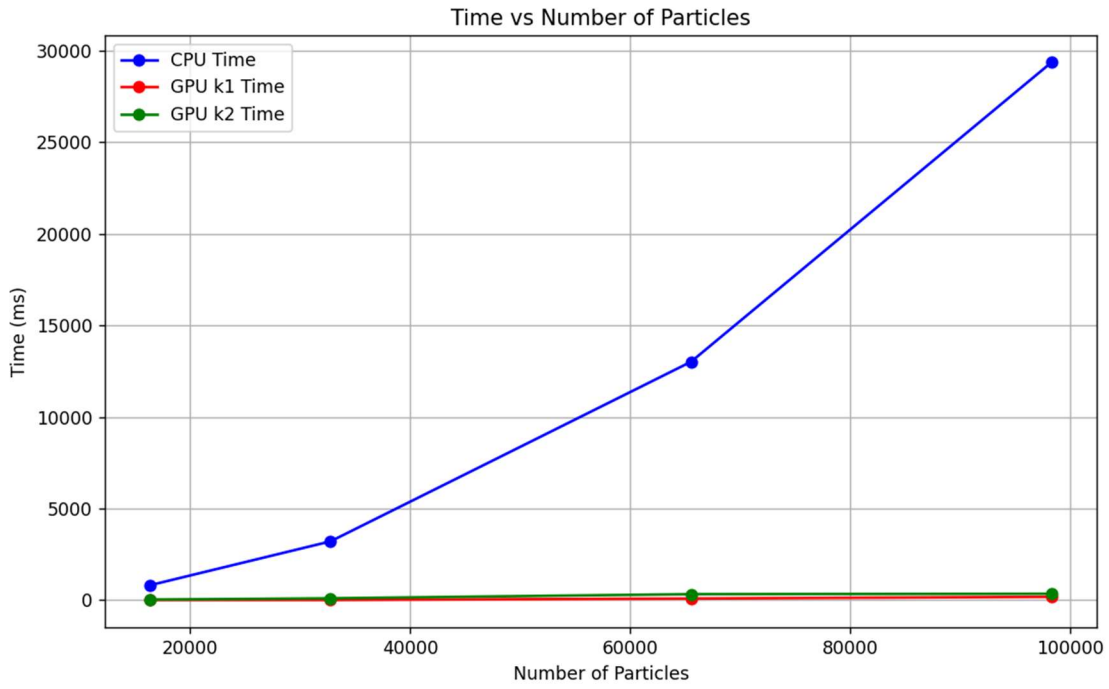
The serial part running on CPU took 30 seconds for 98304 particles. Whereas for the same number of particles GPU kernel with one thread per particle took 183.46662 ms and GPU kernel with one thread per pair of particles took 351.84 ms.

Even for N = 1048576 GPU Kernel 2 took 26 seconds while GPU Kernel 1 took only 3.8 seconds. Beyond this number the operation to calculate number of blocks will cause integer overflow and the program will not run as desired.

Execution times for CPU and GPU with different number of threads.

No. of particles	**CPU (ms)**	**GPU k1 (ms)**	**GPU k2 (ms)**
16384	29350	183.467	351.843
32768	13010	92.124	329.875
65536	3210	13.475	103.584
98304	820	15.136	34.444

The following graph visualizes the relationship between number of particles and time taken by CPU and GPU with different number of threads.



Screenshots:

```
[tpatel44@mcs1 assignment3]$ nvcc -O2 q1.cu -o q1.x
[tpatel44@mcs1 assignment3]$ ./q1.x
------------------------- N  =  98304 -------------------------
----------------------------------------
Serial code on CPU
CPU Time: 29350.0000000 ms
CPU distance: 0.0000084
----------------------------------------
----------------------------------------
One thread per particle
GPU kernel 1 took 183.4666290 ms
GPU kernel 1 distance: 0.0000084
----------------------------------------
----------------------------------------
One thread per pair
GPU kernel 2 took 351.8438721 ms
GPU kernel 2 distance: 0.0000000
----------------------------------------
```

```
[tpatel44@mcs1 assignment3]$ nvcc -O2 q1.cu -o q1.x
[tpatel44@mcs1 assignment3]$ ./q1.x
------------------------- N  =  65536 -------------------------
----------------------------------------
Serial code on CPU
CPU Time: 13010.0000000 ms
CPU distance: 0.0000089
----------------------------------------
----------------------------------------
One thread per particle
GPU kernel 1 took 92.1247711 ms
GPU kernel 1 distance: 0.0000089
----------------------------------------
----------------------------------------
One thread per pair
GPU kernel 2 took 329.8749695 ms
GPU kernel 2 distance: 0.0000089
----------------------------------------
```

```
[tpatel44@mcs1 assignment3]$ nvcc -O2 q1.cu -o q1.x
[tpatel44@mcs1 assignment3]$ ./q1.x
------------------------- N  =  32768 -------------------------
----------------------------------------
Serial code on CPU
CPU Time: 3210.0000000 ms
CPU distance: 0.0000323
----------------------------------------
----------------------------------------
One thread per particle
GPU kernel 1 took 13.4570885 ms
GPU kernel 1 distance: 0.0000323
----------------------------------------
----------------------------------------
One thread per pair
GPU kernel 2 took 103.5844193 ms
GPU kernel 2 distance: 0.0000323
----------------------------------------
```

```
[tpatel44@mcs1 assignment3]$ nvcc -O2 q1.cu -o q1.x
[tpatel44@mcs1 assignment3]$ ./q1.x
------------------------- N  =  16384 -------------------------
----------------------------------------
Serial code on CPU
CPU Time: 820.0000000 ms
CPU distance: 0.0000607
----------------------------------------
----------------------------------------
One thread per particle
GPU kernel 1 took 15.1367998 ms
GPU kernel 1 distance: 0.0000607
----------------------------------------
----------------------------------------
One thread per pair
GPU kernel 2 took 34.4446068 ms
GPU kernel 2 distance: 0.0000607
----------------------------------------
```