**Part A:**

**MaxCol.cu Timings**

|  |  |  |  |
| --- | --- | --- | --- |
| **GPU (TITAN)** | **1024x1024 matrix**  **(ms)** | **2048x2048 matrix**  **(ms)** | **4096x4096 matrix**  **(ms)** |
| **8 threads/block** | .295797 | .888128 | 2.533504 |
| **16 threads/block** | .300395 | .512331 | 1.720810 |
| **32 threads/block** | .301173 | .515307 | .962059 |
| **64 threads/block** | .289120 | .506837 | .960864 |
| **128 threads/block** | .284949 | .510197 | .971157 |

Figure 1.1: The table represents the timing in milliseconds for running the GPU kernel given certain parameters for size of the matrix and threads per a block.

**Part B:**

**minDist.c Timings**

|  |  |
| --- | --- |
| **NUMPARTICLES** | **ELAPSED CPU TIME (ms)** |
| **1024 particles** | 4.754 ms |
| **8192 particles** | 270.710 ms |
| **32768 particles** | 3655.281 ms |

Figure 2.1: The table represents the timing in milliseconds for running the minDist.c given parameters for number of particles.

**minDistSOA.c Timings**

|  |  |
| --- | --- |
| **NUMPARTICLES** | **ELAPSED CPU TIME (ms)** |
| **1024 particles** | 4.76967 ms |
| **8192 particles** | 230.361 ms |
| **32768 particles** | 3675.72 ms |

Figure 2.2: The table represents the timing in milliseconds for running the minDistSOA.c given parameters for number of particles. This program is written using a indexing method similar to creating a struct of array.

**minDistSOAGPU.cu Timings**

|  |  |  |  |
| --- | --- | --- | --- |
| **GPU (TITAN)** | **1024 particles**  **(ms)** | **8192 particles**  **(ms)** | **32768 particles**  **(ms)** |
| **4 threads/block** | 2.23388 ms | 43.7829 ms | 546.471 ms |
| **16 threads/block** | 1.74338 ms | 15.3141 ms | 144.544 ms |
| **64 threads/block** | 1.67739 ms | 6.39443 ms | 65.4745 ms |

Figure 2.3: The table represents the timing in milliseconds for running the minDistSOAGPU.cu given parameters for number of particles and number of threads/block. This was simulated on the GPU titan.