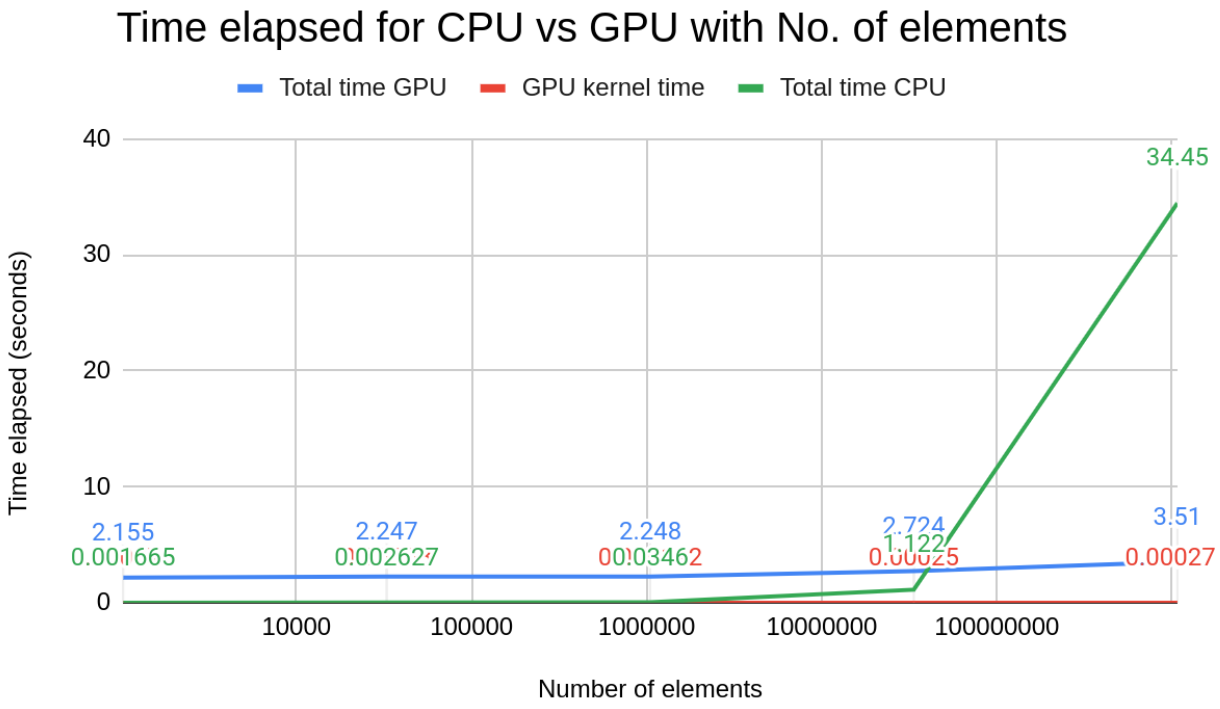
Question 1

The platform used to run experiments for this question is the **Discovery Cluster** using sbatch. More information on the code and how to run is provided in the Readme and in the log file.

In the below table, time elapsed by CPU and GPU implementation of histogram binning are shown with increase in the number of elements. The grid size and block size are chosen such that the total number of threads is equal to the total number of elements for maximum concurrency.

Number of classes = 10
Number of threads set in CPU = 64

Number of elements	Total threads = grid_size * block_size	Total time elapsed GPU (seconds)	Time elapsed by CUDA kernel (seconds)	Time elapsed by CPU (seconds)
1024 (2^10)	(32, 1, 1) * (32, 1, 1)	2.155	0.000079	0.001665
32768 (2^15)	(64, 1, 1) * (512, 1, 1)	2.247	0.000080	0.002627
1048576 (2^20)	(1024, 1, 1) * (1024, 1, 1)	2.248	0.000192	0.0346
33554432 (2^25)	(2048, 1, 1) * (1024, 1, 1)	2.724	0.000250	1.122
1073741824 (2^30)	(2048, 1, 1) * (1024, 1, 1)	3.510	0.000271	34.45



- From the above graph, the time taken to copy data from host (cpu memory) to device (gpu memory) and vice versa is significantly higher compared to the computation time taken by the CUDA kernel.
- With the increase in the number of elements for binning, the time taken by CPU OpenMP implementation significantly increases. Whereas, the time taken by the CUDA kernel increases comparatively slowly.
- But with the increase in the number of elements for binning, the time taken by the host → device and device → host communication increases considerably.

In the below table, time elapsed by CPU and GPU implementation of histogram binning are shown with varying number of classes.

Number of elements = 32768
Number of threads set in CPU = 64

Number of classes	Total time elapsed GPU (seconds)	Time elapsed by CUDA kernel (seconds)	Time elapsed by CPU (seconds)
1	2.21	0.000081	0.0019
10	2.27	0.000085	0.0026

20	2.23	0.000087	0.0031
40	2.26	0.000084	0.0039
80	2.28	0.000072	0.0055

- From the above table, the time taken by the GPU implementation and the time taken by the CUDA kernel remains approximately the same as we increase the number of classes.
- Whereas, the time taken by the CPU OpenMP implementation increases quite significantly as we increase the number of classes.

In the below table, time elapsed by GPU implementation of histogram binning are shown with increase in grid size.

Total threads = grid_size * block_size	Total time elapsed GPU (seconds)	Time elapsed by CUDA kernel (seconds)
(32, 1, 1) * (128, 1, 1)	2.22	0.000084
(64, 1, 1) * (128, 1, 1)	2.345	0.000080
(128, 1, 1) * (128, 1, 1)	2.26	0.000080
(256, 1, 1) * (128, 1, 1)	2.1543	0.000080
(512, 1, 1) * (128, 1, 1)	2.349	0.000089
(1024, 1, 1) * (128, 1, 1)	2.346	0.000116

- From the above table, the time taken by the GPU implementation and the time taken by the CUDA kernel remains approximately the same as we increase the grid size.
- There is a slight increase in the time elapsed by the CUDA kernel when the grid size = 1024.