Project 5
Diego Loya

Fractal

Fractal runtimes using double		
# of frames	Width	Runtime (secs)
30	250	0.0093
60	250	0.0174
30	500	0.0312
60	500	0.0622

Fractal runtimes using float		
# of frames	Width	Runtime (secs)
30	250	0.0055
60	250	0.0103
30	500	0.0190
60	500	0.0375

- We can see that when using float, performance increases considerably. This is because CUDA converts doubles into floats by default in lower versions. It may also run faster since it is using a less accurate path.
- The time is much faster than OpenMP because we are able to parallelize individual pixels without any overlap, which results in a much faster runtime.
- We get the following error: "FATAL: Module nvidia not found. Could not allocate memory. This is because the login nodes don't have access to the GPU.

TSP

TSP runtimes	
# of cities	Best Runtime (secs)
1291	1.5089
1379	1.7758
1400	1.6964

- The times for all three different cities were very similar. The reason for this is that most of the time consumption is from transfer between host and device, which does not change much between the problem sizes.
- The time for 1291 cities is much slower in CUDA because there is a lot of communication going on between the GPU and the CPU every time a kernel is called. This affects the performance. Also, the way the program runs, there is a lot of overlap.

- Because when doing ‘(j>=i+2)’, all elements in the first block and more, will fail and therefore will not evaluate any edge pair.