

Programming Assignment 4: Histogramming

Due Date: October 3, 2023 at 11:59 PM

Note: This programming assignment is required for ALL students.

1. Objective

The purpose of this programming assignment is to implement an efficient histogramming algorithm for an input array of integers within a given range. Each integer will map into a single bin, so the values will range from 0 to (#bins - 1). The histogram bins will use unsigned 8-bit counters that must be saturated at 255 (i.e., rolling back to 0 is NOT allowed).

2. Procedure

Step 1: Download the programming assignment 4 materials from blackboard to your home folder at Karpenski. Unzip it.

```
unzip p4-histogram.assignment.zip
```

Step 2: Edit `kernel.cu` to implement host and device kernel code for the histogram. Remember that the resulting 8-bit counters must be saturated at 255. Begin with a naïve implementation then optimize it gradually. Keep a journal of every optimization you tried including the ones you abandoned because they limited you or worsened performance. This journal will be included in your report for this programming assignment.

Step 3: Compile and test your code.

```
make  
  
./histogram          # Uses default input size and default number of bins  
./histogram <m>      # Uses input of size m and default number of bins  
./histogram <m> <n>   # Uses input of size m and bin count of n
```

Step 4: Create a report in a new file named `report.txt` that describes all optimizations you tried regardless of whether you committed to them or abandoned them and whether they improved or hurt performance. For each optimization, include in your report:

- A description of the optimization
- Any difficulties you had with completing the optimization correctly
- The change in execution time after the optimization was applied
- An explanation of why you think the optimization helped or hurt performance

Step 5: Submit your assignment. You should only submit the following files:

- `kernel.cu`
- `report.txt`

Compress the files and name them after your last name like so:

```
tar -cvf p4_<your last name>.tar kernel.cu report.txt
```

If the last name is alan, then the tar file name is `p4_alan.tar`

Submit the tar file in blackboard.

3. Grading:

Your submission will be graded based on the following criteria.

- Functionality: 90 points
 - Correct code and output results
 - Correct handling of boundary cases
 - Check return values of all CUDA APIs
- Report: 10 points
 - Effort and thoughtfulness put into optimizing the algorithm/code
 - Completeness and accuracy of the report
 - Neatness and clarity