# Lab 6: Histogramming

Due Date: …

## 1. Objective

The purpose of this lab 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. no roll back to 0 allowed).

## 2. Procedure

**Step 1:** Update your local repository to obtain the code needed for the assignment.

cd <labs-directory>

hg pull

hg update

**Step 2:** Edit lab6/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 lab.

**Step 3:** Compile and test your code.

cd lab6

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 <lab-directory>/report.txt which describes of 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 student id like so:

tar -cf id.tar kernel.cu report.txt

Send the compressed folder by email to <TA’s email address> with “ECE408 Lab 6” in the subject line. Submissions with incorrect subject lines may not be processed.

**3. Grading:**

Your submission will be graded based on the following criteria.

* Functionality: 40%
  + Correct code and output results
  + Correct handling of boundary cases
* Optimization effort: 25%
  + Effort and thoughtfulness put into optimizing the algorithm/code
* Report: 35%
  + Completeness and accuracy of the report
  + Neatness and clarity