

CS344 Introduction to Parallel Programming

Lesson 3: Fundamental GPU Algorithms (Reduce, Scan, Histogram)

L3-3.2-Fundamental GPU Algorithms

LECTURE 3

FUNDAMENTAL GPU ALGORITHMS

- REDUCE
- SCAN
- HISTOGRAM

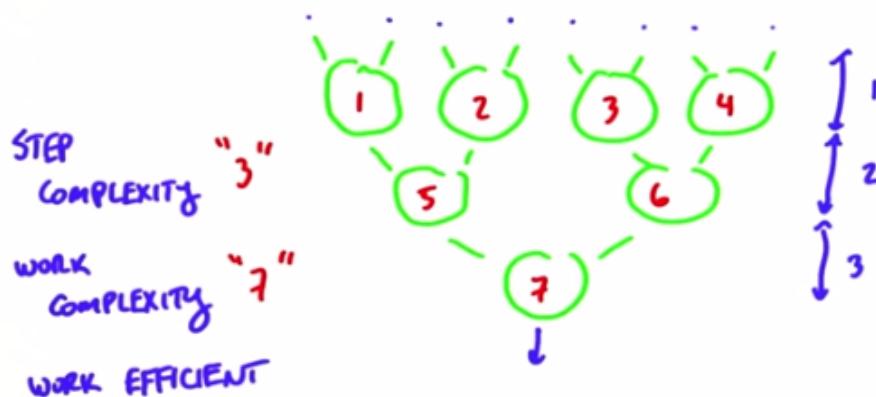


L3-3.3-Digging Holes Again

DIGGING HOLES AGAIN



L3-3.4-Steps and Work



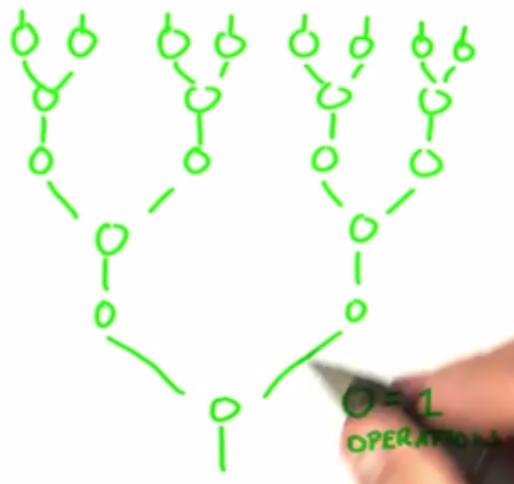
L3-3.5-A Quiz on Step And Work

QUIZ

#STEPS?



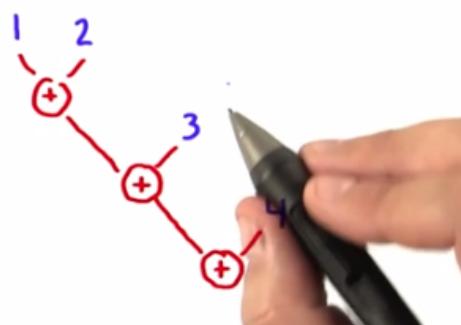
TOTAL AMOUNT OF WORK?



L3-3.6-Reduce Part 1

REDUCE

$$1 + 2 + 3 + 4 + \dots$$

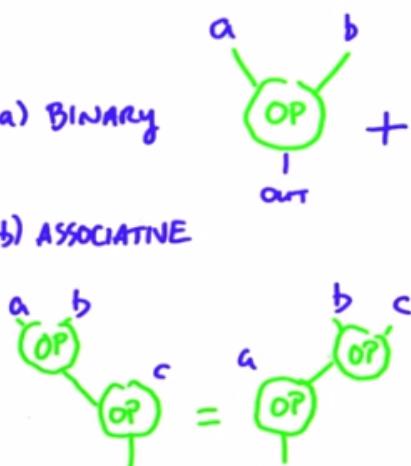


L3-3.7-Reduce Part 2

REDUCE: INPUTS

- 1) SET OF ELEMENTS
- 2) REDUCTION OPERATOR $\overbrace{\quad}$

- $\left\{ \begin{array}{l} \text{a) BINARY} \\ \text{b) ASSOCIATIVE} \end{array} \right.$



L3-3.8-Binary and Associative Operators

QUIZ

CHECK THE OPERATIONS THAT ARE
BOTH BINARY AND ASSOCIATIVE.

- | | |
|---|---|
| <input type="checkbox"/> MULTIPLY ($a * b$) | <input type="checkbox"/> BITWISE AND ($a \& b$) |
| <input type="checkbox"/> MINIMUM ($a \min b$) | <input type="checkbox"/> EXPONENTIATION (a^b) |
| <input type="checkbox"/> FACTORIAL ($a!$) | <input type="checkbox"/> DIVISION ($\frac{a}{b}$) |
| <input type="checkbox"/> LOGICAL OR ($a \parallel b$) | |



L3-3.9-Serial Implementation of Reduce

SERIAL IMPLEMENTATION OF REDUCE

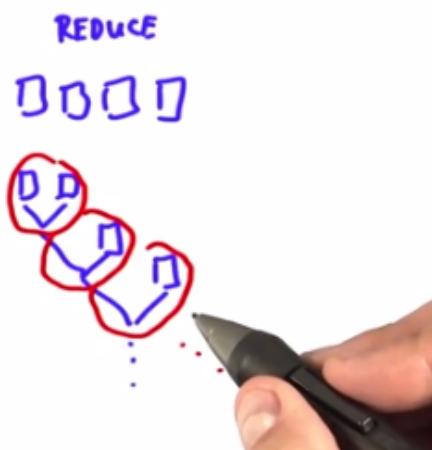


SERIAL IMPLEMENTATION OF REDUCE

SERIAL CODE

```

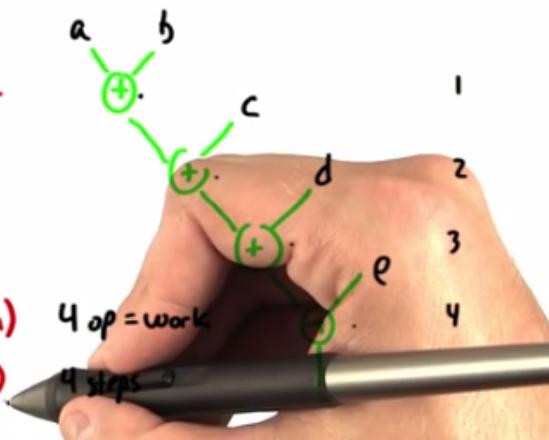
sum = 0
for (i=0; i<elts.len(); i++) {
    sum = sum + elts[i]
}
return sum
  
```



QUIZ

Which are true about a serial reduce code running on an input of size n ?

- It takes n operations.
- It takes $n-1$ operations.
- Its work complexity is $O(n)$
- Its step complexity is $O(1)$



L3-3.10-Parallel Reduce

QUIZ

How do you rewrite

$$(a+b)+c+d$$

to allow parallel execution?

(Use paren's to show grouping.)

L3-3.11-Step Complexity of Parallel Reduce

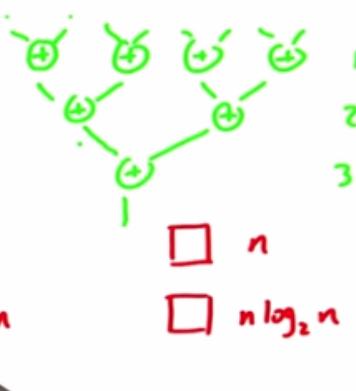
STEP COMPLEXITY OF PARALLEL REDUCTION

N	STEPS
2	1
4	2
8	3
...	

QUIZ

\sqrt{n}

$\log_2 n$

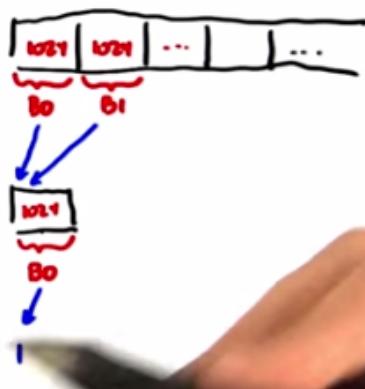


Brent's Theorem

L3-3.12-Reduction Using Global and Shared Memory

REDUCING 1M ELEMENTS

(1) 1024 BLOCKS x 1024 THREADS



(2) 1 BLOCK x 1024 THREADS

Code available

<https://github.com/udacity/cs344/tree/master/Unit3%20Code%20Snippets>

SHARED VS GLOBAL MEMORY BANDWIDTH

THE GLOBAL MEMORY VERSION USES



TIMES AS MUCH GLOBAL MEM BW AS
THE SHARED MEM VERSION?

L3-3.13-Scan

SCAN

— EXAMPLE

INPUT: 1 2 3 4

OPERATION: ADD

OUTPUT: 1 3 6 10

- ADDRESSES SET OF PROBLEMS OTHERWISE DIFFICULT TO PARALLELIZE
- NOT USEFUL IN SERIAL WORLD BUT VERY USEFUL IN PARALLEL
- TODAY: EXPLAINING WHAT + HOW
BUT NOT WHY (NEXT LECTURE)

L3-3.14-Balancing Checkbook With Scan



TRANSACTION	BALANCE
\$ 20	20
5	25
- 11	14
- 9	5
- 3	2
15	17
INPUT	OUTPUT

L3-3.15-Inputs to Scan

INPUTS TO SCAN

- INPUT ARRAY
- BINARY ASSOCIATIVE OPERATOR } LIKE REDUCE
- IDENTITY ELEMENT [$I \text{ op } a = a$]

OP	I	BECAUSE
+	\emptyset	$\emptyset + a = a$
min (on unsigned chars)	FF	$\min(\text{FF}, a) = a$

QUIZ

WHAT IS THE IDENTITY FOR ...

Multiply

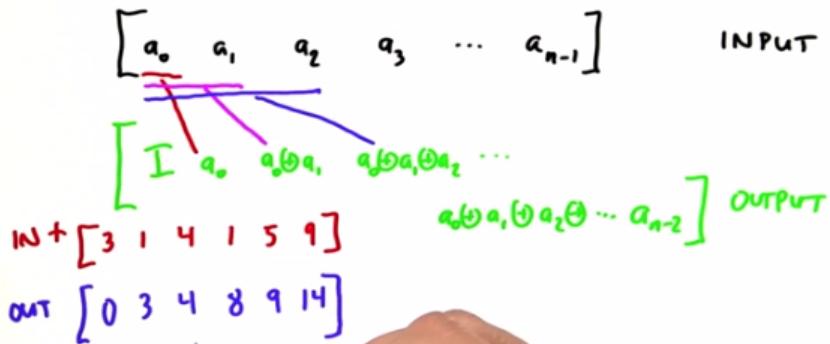
Logical or

Logical and

L3-3.16-What Scan Actually Does

What Scan Does

Input: array A, operator \oplus , identity I
PLUS \oplus



QUIZ

MAX-SCAN
ON UNSIGNED
INTS

$[3 \ 1 \ 4 \ 1 \ 5 \ 9]$ IDENTITY?

OUTPUT?

$[\square \ \square \ \square \ \square \ \square \ \square]$

L3-3.17-Why Do We Care About Parallel Scan

SERIAL IMPLEMENTATION OF SCAN

```

int acc = identity;
for (i=0; i<elements.length(); i++) {
    acc = acc op element[i];
    out[i] = acc;
}

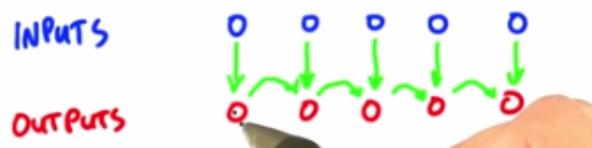
```

INCLUSIVE

WORK? n

STEPS? n

WHY SCAN IS USEFUL FOR PARALLELIZATION



L3-3.18-Inclusive vs. Exclusive Scan

INCLUSIVE VS EXCLUSIVE SCAN

INPUT: $[13 \ 7 \ 16 \ 21 \ 8 \ 20 \ 13 \ 12]$

EXCLUSIVE SCAN OUTPUT: $[0 \ 13 \ 20 \ 36 \ 57 \ 65 \ 85 \ 98]$ OUTPUT: ALL ELEMENTS BEFORE, NOT CURRENT ELT.

INCLUSIVE SCAN OUTPUT: $[13 \ 20 \ 36 \ 57 \ 65 \ 85 \ 98 \ 110]$ OUTPUT: ALL ELEMENTS BEFORE AND CURRENT ELT.

L3-3.19-Serial Implementation of Scan

SERIAL IMPLEMENTATION OF SCAN

```
int acc = identity;
for (i=0 ; i<elements.length(); i++) {
    acc = acc op element[i];
    out[i] = acc;
}
```

QUIZ: CONVERT
TO EXCLUSIVE
SCAN.

L3-3.20-Inclusive Scan Revisited

INCLUSIVE SCAN EXAMPLE, REVISITED

IN: $[3 \ 1 \ 4 \ 1 \ 5 \ 9]$

 OUT: $[3 \ 4 \ 8 \ 9 \ 14 \ 23]$

QUIZ	CONSTANT $O(1)$	LINEAR $O(n)$	$O(n^2)$
STEPS?	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
WORK?	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

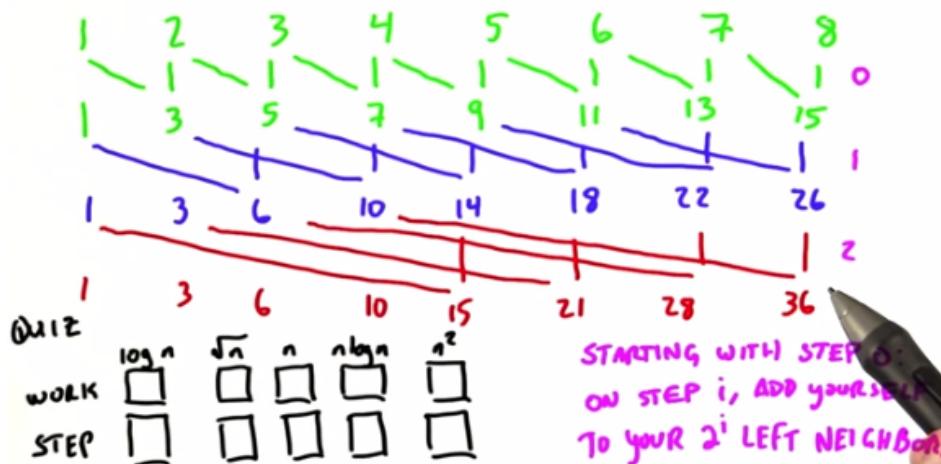
L3-3.21-Hillis Steele vs. Blelloch Scan

TWO PARALLEL SCAN ALGORITHMS



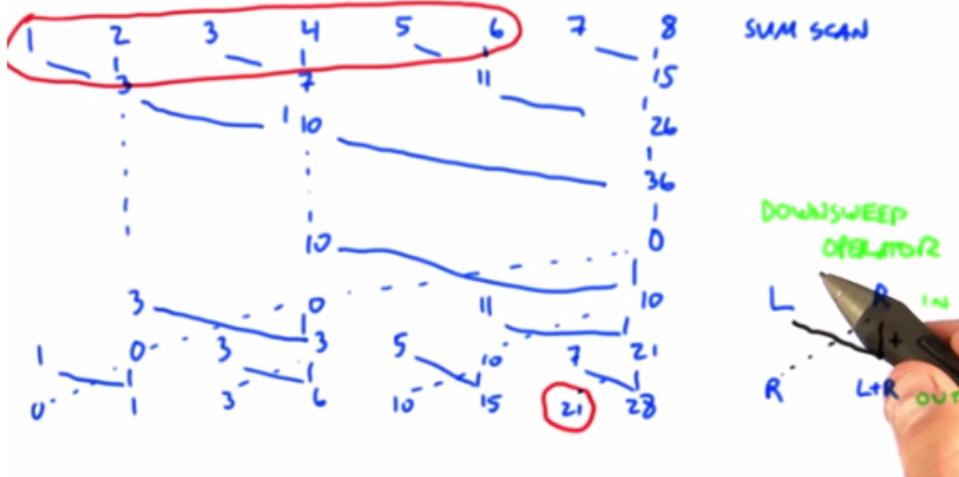
L3-3.22-Hillis Steele Scan

HILLIS/STEELE INCLUSIVE SCAN



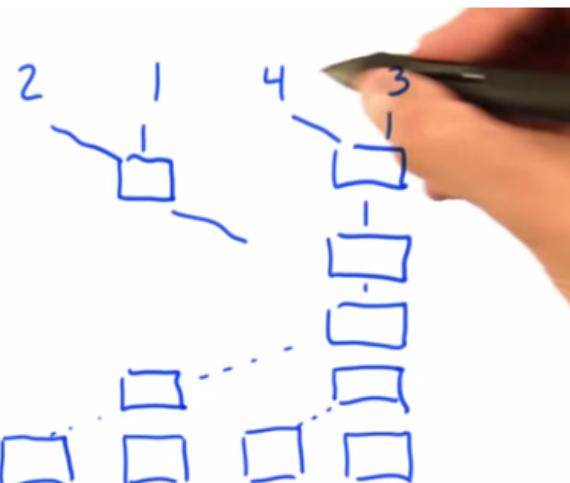
L3-3.23-Blelloch Scan

BLELLOCH SCAN · REDUCE/DOWNSWEEP · EXCLUSIVE



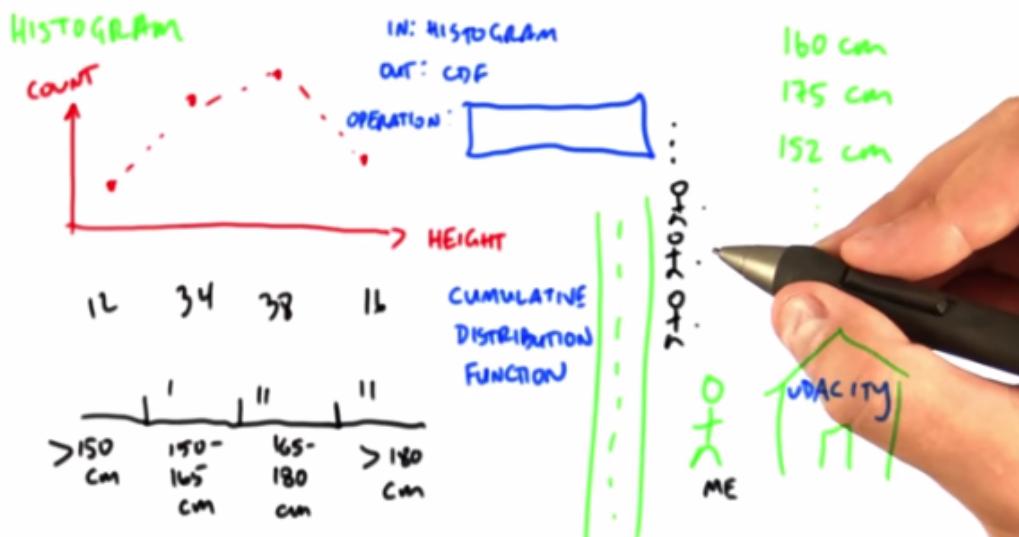
① UNIT

MAX SCAN
USING
REDUCE/
DOWNSWEEP





L3-3.26-Histogram



L3-3.27-Serial Implementation of Histogram

SERIAL ALGORITHM : HISTOGRAM

```

for (i=0; i<BIN-COUNT; i++)
    result[i] = 0;
for (i=0; i<BIN-COUNT; i++)
    result[computeBin(measurements[i])]++;
    ↑ TO WHICH BIN DOES THIS MEASUREMENT BELONG?
  
```

INPUT:
 155
 150
 175
 170

0	<150
2	150-165
2	165-180
0	>180

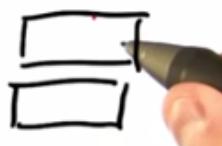
Q112

n measurements

b bins

MAXIMUM # OF MEASUREMENTS/BIN

AVERAGE # OF MEASUREMENTS/BIN



L3-3.28-Parallel Implementation of Histogram

SERIAL ALGORITHM: HISTOGRAM

```

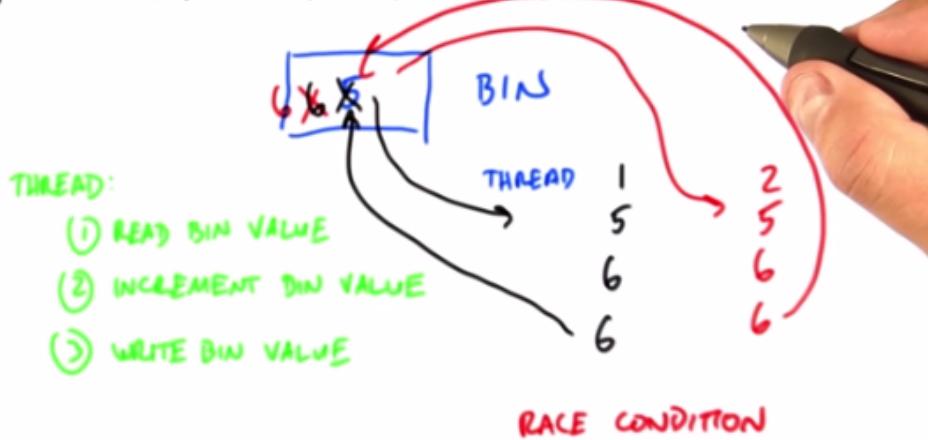
for (i=0; i< BIN_COUNT; i++)
    result[i] = 0;
for (i=0; i< measurements.size(); i++)
    result[computeBin(measurements[i])]++;

```

Code Available

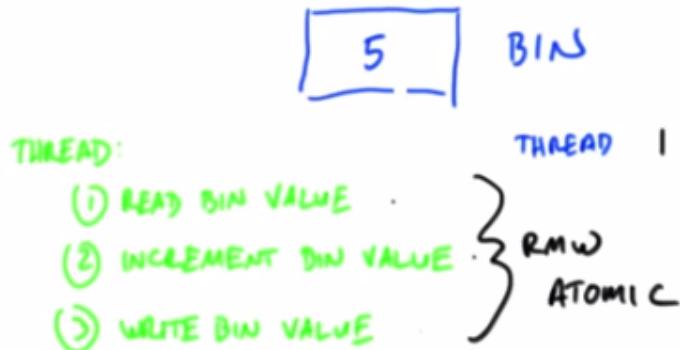
<https://github.com/udacity/cs344/blob/master/Unit3%20Code%20Snippets/histo.cu>

WHY THE OBVIOUS METHOD DOESN'T WORK



L3-3.29-Implementing Histogram Using Atomics

METHOD 1: ACCUMULATE USING ATOMICS



Code Available

<https://github.com/udacity/cs344/blob/master/Unit3%20Code%20Snippets/histo.cu>

QUIZ

- Histogram with 1M elements
- You can choose # of bins



L3-3.30-Implementing Histogram Using Local Memory

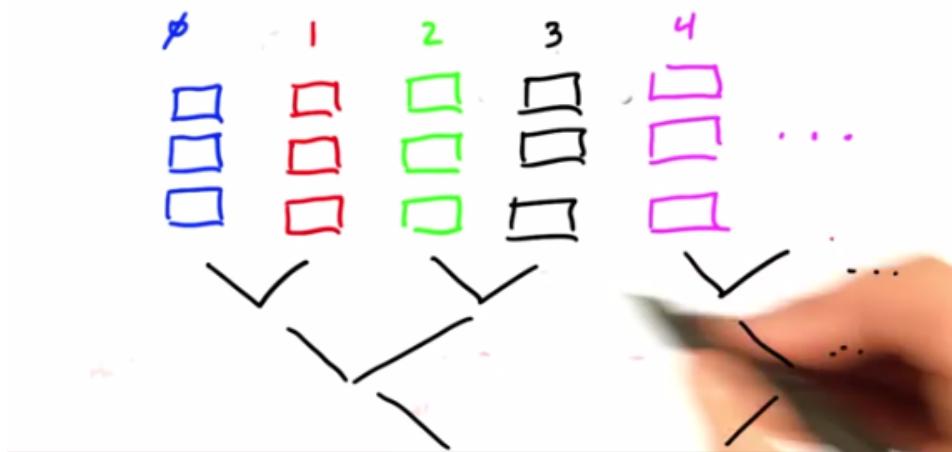
PER-THREAD PRIVATIZED (LOCAL) HISTOGRAMS, THEN REDUCE

128 ITEMS · 8 THREADS · 3 BINS
(EACH THREAD GETS 16 ITEMS)



L3-3.31-Calculating Global Histogram Using Reduction

REDUCING 8 LOCAL HISTOGRAMS



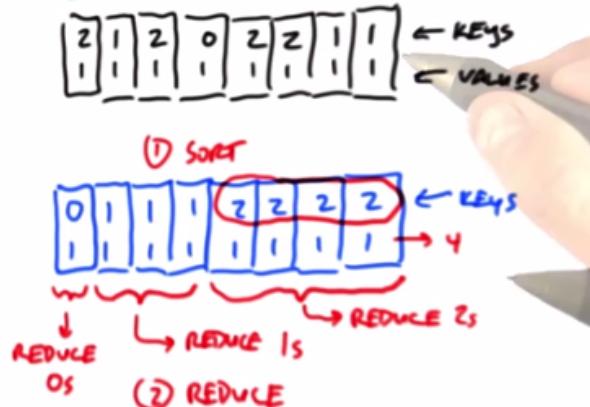
L3-3.32-Sort Then Reduce By Key

SORT, THEN REDUCE by key

8 ENTRIES

3 BINS

(0,1,2)



L3-3.33-What Operation Should We Use

PER-THREAD PRIVATIZED (LOCAL) HISTOGRAMS, THEN REDUCE

128 ITEMS · 8 THREADS · 3 BINS

(EACH THREAD GETS 16 ITEMS)



HOW CAN WE COMBINE THESE 8 LOCAL HISTOGRAMS INTO ONE GLOBAL HISTOGRAM?

L3-3.34-Final Thoughts on Histogram

FINAL THOUGHTS ON HISTOGRAM

- ATOMICS
- PER-THREAD HISTOGRAMS, THEN REDUCE (1)
- SORT, THEN REDUCE BY KEY (2)

256 THREADS, 8 BINS:

HOW MANY ATOMIC ADDS?

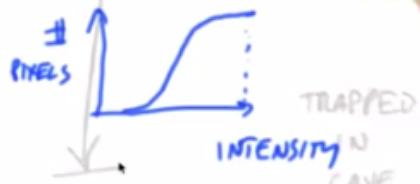
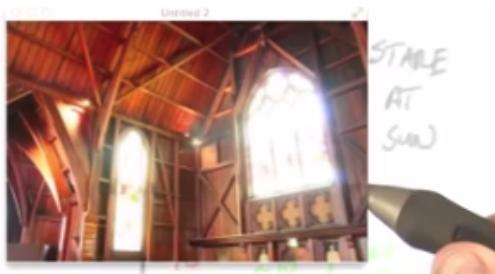
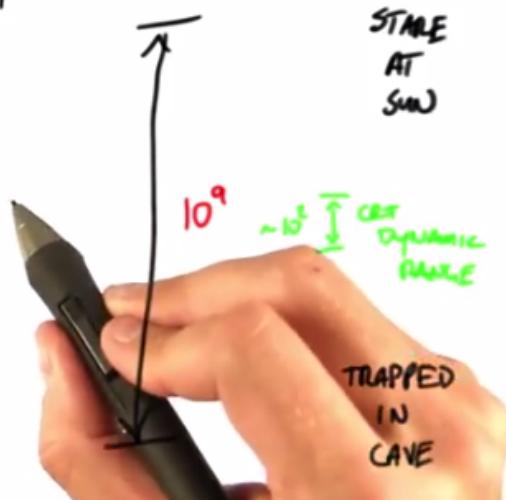
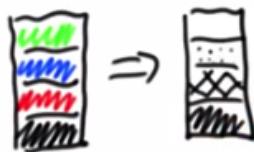
ATOMIC TECHNIQUE: □

REDUCE TO 8-ELEMENT HISTOGRAM THEN ATOMICS □

L3-3.35-Description of Problem Set #3

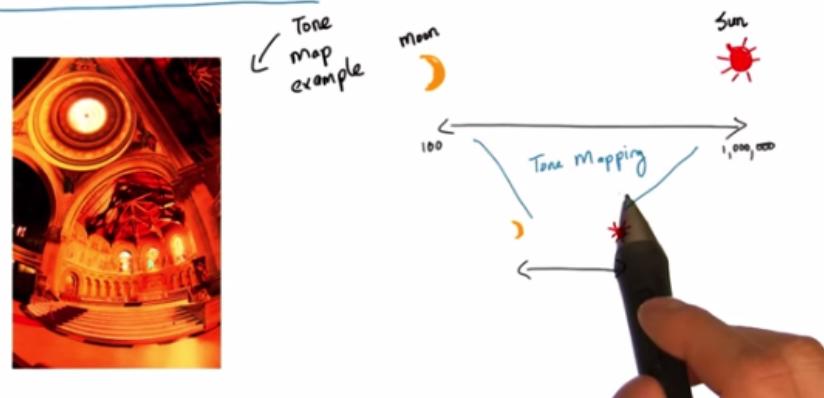
DESCRIPTION OF ASSIGNMENT

TONE MAPPING

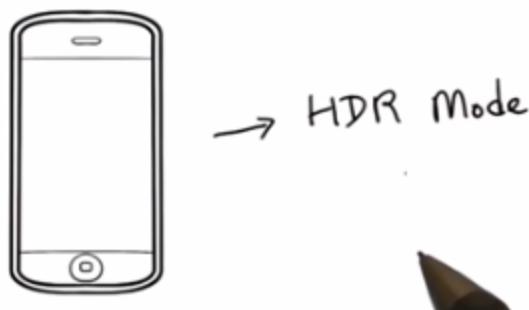


L3-3.37-Problem Set #3

Problem Set #3



Problem Set #3



Problem Set #3

Histogram Equalization

- ① Map ✓
- ② Reduce ✓
- ③ Scatter ✓
- ④ Scan ✓

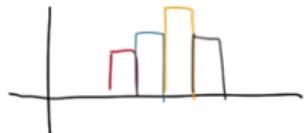
Problem Set #3

Main Idea !

- ① Histogram of the brightness values
- ② A scan of the histogram using the \oplus

Problem Set #3

① Minimum brightness



② Maximum brightness

③ Parallel Reduce

Problem Set #3

