Module 7 Lab: Reduction
Answers

Sajina Pathak

Question 1

How many times does a single thread block synchronize to reduce its portion of the array to a single value?

A single thread block synchronizes $\log_2(\text{BLOCK\_SIZE})$ times.
For example, if BLOCK_SIZE = 512, then:

$$\log_2(512) = 9$$

So, it will synchronize **9 times**.

Working Example

Suppose:

$$\text{BLOCK\_SIZE} = 8$$

Input array:

$$[A_0, A_1, A_2, A_3, A_4, A_5, A_6, A_7]$$

Each thread handles two elements:

- Thread 0 handles → `[in[0], in[4]]` → sum = $1 + 5 = 6$
- Thread 1 handles → `[in[1], in[5]]` → sum = $2 + 6 = 8$
- Thread 2 handles → `[in[2], in[6]]` → sum = $3 + 7 = 10$
- Thread 3 handles → `[in[3], in[7]]` → sum = $4 + 8 = 12$

Question 2

What is the minimum, maximum, and average number of "real" operations that a thread will perform?

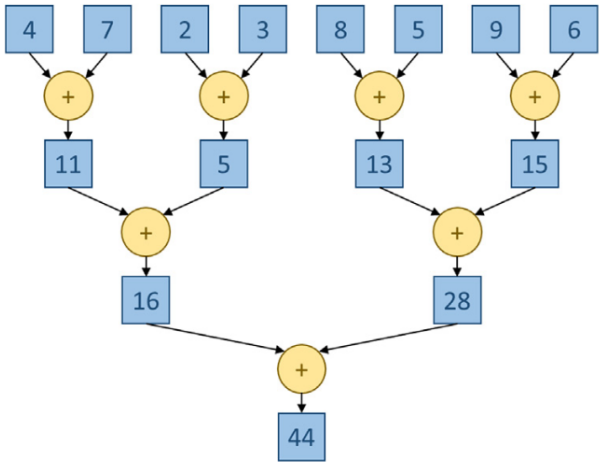

Figure 1: A Parallel sum of Reduction Tree

- **Minimum**: 0 real operations (some threads do not participate in any reductions).
- **Maximum**: $\log_2$(BLOCK_SIZE) real operations.
- **Average**: Roughly half of $\log_2$(BLOCK_SIZE) real operations.

For BLOCK_SIZE = 512:

$$\log_2(512) = 9$$

- **Minimum**: 0 operations.
- **Maximum**: 9 operations.
- **Average**: $\approx$ 4 to 5 operations.