

ECS223a Parallel Algorithms

Homework 1

You are encouraged to talk to other people about these problems, but please **write up the solutions by yourself**. Cite any conversations you had with others, as well as books, papers or Web sites you consulted.

Always explain the answer in **your own words**; do not copy text from anywhere. Explain your solution as you would to someone who does not understand it, for instance to a beginning graduate student or an advanced undergraduate. Do not give several solutions to one problem; pick the best one and explain it.

Please type your homework and submit it using SmartSite. If you know LaTeX, use that. If you don't know it, this might be a good time to learn. Include pictures if appropriate; you can scan them in or produce them with a computer sketching program.

1. Do exercise 1.3 in Blelloch's chapter on scan operations, which we read for the class on 10/1 (see the Web page). The problem is to construct an adder using prefix-sum. Please do not construct the adder by converting the bit strings into integers and then adding the integers.
2. Do problem 3.5 in the book (write the algorithm for putting removed nodes back into the linked-list parallel prefix result). Look at the Algorithm 3.1.2 and use similar variable names and pseudo-code as much as you can. Your algorithm should have work $O(\lg n)$. What is the best depth you can achieve? $O(\lg n \lg \lg n)$, $O(\lg n)$, $O(\lg \lg n)$?
3. Do Exercise 2.44 at the end of Chapter 2. Here is an example of input and output, for $n = 16$ (so there are four "colors", 0-3):

```
colors = [2, 0, 0, 1, 3, 3, 2, 3, 1, 2, 0, 1, 3, 0, 2, 0]
values = [1, 1, 2, 2, 5, 1, 2, 1, 4, 3, 3, 2, 0, 2, 3, 1]
output = [1, 1, 3, 2, 5, 6, 3, 7, 6, 6, 6, 9, 7, 8, 9, 9]
```

Hint: If the length of the list were $O(\lg n)$, what would the sequential running time be?

4. Do Exercise 2.45 at the end of Chapter 2, which uses the result of 2.44 to speed up radix sort when the numbers are all small.