Parallel and Concurrent Algorithms (15-852), Spring 2024
Assignment #1 Due: Thu, Feb 1

Problem 1: Finding the Max

In this problem you need to describe an algorithm in the MP-RAM with arbitrary-way forking for finding the maximum of a set of n values.

This is split into four parts with different bounds.

1. (5pt) Describe an $O(n^2)$ work and constant span algorithm.
2. (10pt). Using the result from the first part, describe a recursive algorithm that runs in $O(n)$ and $O(\log\log n)$ span. You might find the sorts of techniques used for the $O(\log\log n)$ merging algorithm helpful.
3. (10pt). Describe an $O(n^{3/2})$ work and constant span algorithm using ideas from the previous parts.
4. (10pt). Describe an $O(n)$ work and constant span randomized algorithm. Hint: use sampling to first reduce the problem down to e.g. $O(n^{7/8})$ size, and use something like the algorithm from the previous part on the sample.

Problem 2: Union Scan

You are given a sequence of keys, and you want to insert them into a set one by one, while keeping all the intermediate sets. Lets call this the all prefix-sets (APS) problem.

Assume you have a library for sets that supports insertion in $O(\log n)$ work and span; and it supports union of two sets in $O(m\log(n/m))$ work and $O(\log n)$ span where m is the size of the smaller set and n the size of the larger. Both these operations generate their result without destroying their input.

Using union you can implement the APS problem by inserting each element into its own singleton set (in parallel), and then doing a scan (prefix-sum) using union as the associative combining function. This gives the correct answer since for each position it will return the union of all previous elements.

1. (10pt). Determine the work and span of the scan algorithm in the notes when used with union. Note that it will not be $O(n)$ work since the union does not take $O(1)$ work.

2. (10pt). Now consider the following modification of the prefix sum algorithm. First break the input into $\sqrt{n}$ blocks of size $\sqrt{n}$ each. Now take the sum (based on the associative function) of each of the blocks, in parallel across the blocks, but sequentially within each block. Then run a scan across the $\sqrt{n}$ partial sums sequentially. Finally use the results of the partial sums to fill in the final results within each chunk, again in parallel across the blocks, but sequentially within the blocks. Below is an example after each step using addition as the combining function.

   Break into blocks: $[[2, 3, 1, 2], [1, 3, 1, 5], [2, 7, 1, 3], [2, 1, 3, 1]]$

   Sum within blocks (parallel across blocks, sequential within): $[8, 10, 13, 7]$

   Scan across block sums (sequential): $[0, 8, 18, 31]$

   Scan within blocks starting with the offsets from previous step (parallel across blocks, sequential within):

   $$[0, 2, 5, 6], [8, 9, 12, 13], [18, 20, 27, 28], [31, 33, 34, 37]$$

   Now determine the work and span of this modified scan algorithm when used with union.