Parallel Algorithms	Spring 2022

Problem Set 3

Prof. Nodari Sitchinava — *Due: Wednesday, February 16, 2022 at 9am*

You may discuss the problems with your classmates, however **you must write up the solutions on your own** and **list the names** of every person with whom you discussed each problem.

Start **every** problem on a separate page. ***Any problem submitted by 11:59pm Friday February 11, 2022 will receive an additional 10% of the score you receive on that problem.***

1 Segmented Filter (50 pts)

Design an algorithm to implement segmented filter discussed in class. The input to your algorithm should consist of the following:

- array $A[1..n]$ of numbers that should be filtered,
- array $segs[1..n]$ of bits identifying the starting indices of the segments ($segs[i] = 1$ identifies the start of a segment),
- array $flags[1..n]$ that defines how to filter the elements of the array A.

Your algorithm should:

- Place every item $A[i]$ marked with $flags[i] = 1$ to the left of any item $A[j]$ marked with $flags[j] = 0$ ***within its segment*** – the items should not be moved outside its original segment. Your filtering should be stable, meaning, the order of the items marked with the same $flags$ value should stay the same as in the original input.
- Return an array $k[1..n]$, where for every index i that is the start of a segment (i.e., $segs[i] = 1$), $k[i]$ should store the number of elements marked with $flags[i] = 1$ within that segment (the values of the rest of k do not matter and are up to you).
- Run in $O(\log n)$ time and $O(n)$ work.

For example, given the following input:

$$\begin{aligned} A &= [10\ \ 12\ \ 16\ \ 25\ \ 5\ \ 4\ \ 8\ \ 7\ \ 19\ \ 6\ \ 18] \\ segs &= [1\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 1\ \ 0\ \ 0\ \ 0\ \ 0] \\ flags &= [1\ \ 0\ \ 0\ \ 0\ \ 1\ \ 1\ \ 1\ \ 1\ \ 0\ \ 1\ \ 0] \end{aligned}$$

Your algorithm should modify A in-place to the following and return the following k:

$$\begin{aligned} A &= [10\ \ 5\ \ 4\ \ 12\ \ 16\ \ 25\ \ 8\ \ 7\ \ 6\ \ 19\ \ 18] \\ k &= [3\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 3\ \ 0\ \ 0\ \ 0\ \ 0] \end{aligned}$$

Write down the pseudocode for SEG-FILTER($A[1..n], segs[1..n], flags[1..n]$), prove the correctness of your algorithm and analyze its time and work complexities.

Hint: Think about how the indices of the segment boundaries affect the destination indices of the filtering process.

2 Solving Recurrences (50 pts)

(a) **(25 pts)** Consider the following recurrence:

$$T(n) = \begin{cases} T(\sqrt{n}) + c & \text{if } n > 2 \\ c & \text{if } n \leq 2 \end{cases}$$

Using induction/substitution method prove that the above recurrence solves to $T(n) = O(\log \log n)$.

Hint 1: Avoid falling into the pitfall described on page 86 of CLRS.

Hint 2: If you are having difficulty with the base case, read Section 4.3 of CLRS.

(b) **(25 pts)** Consider the following recurrence:

$$T(n) = \begin{cases} \sqrt{n} \cdot T(\sqrt{n}) + cn & \text{if } n > 2 \\ c & \text{if } n \leq 2 \end{cases}$$

Using induction/substitution method prove that the above recurrence solves to $T(n) = O(n \log \log n)$.

Hint 1: Avoid falling into the pitfall described on page 86 of CLRS.

Hint 2: If you are having difficulty with the base case, read Section 4.3 of CLRS.