ICS 691: Parallel Algorithms
In-class collaborative assignment

Due: In lecture on Tuesday, September 16

Instructions: Break up into two groups of 2-3 students and solve the following problems working on the problems as a group. Write up your solutions and turn them in at the beginning of lecture on Tuesday, September 16. Although you solve the problems as a group, you must write up the solutions individually. For each problem you must write the names of the students who were part of the group within which you discussed that problem. You may work on different problems in different groups, if you like.

1 Recurrences

Solve the following recurrences. Show your work. That is write out rigorous mathematical proof for each solution. For problems (a)-(c), use 2 different methods to prove your solution. If you use Master Theorem for any of the solutions, show why it applies. Use induction to prove at least one of the recurrences.

(a) $T(n) = \begin{cases} T(n/2) + O(1) & \text{if } n > 1 \\ O(1) & \text{otherwise.} \end{cases}$

(b) $T(n) = \begin{cases} T(n/2) + O(1) & \text{if } n > 1 \\ O(n) & \text{otherwise.} \end{cases}$

(c) $T(n) = \begin{cases} T(n/2) + O(n) & \text{if } n > 1 \\ O(1) & \text{otherwise.} \end{cases}$

(d) $T(n) = \begin{cases} T(n/2) + O(\log n) & \text{if } n > 1 \\ O(1) & \text{otherwise.} \end{cases}$

(e) $T(n) = \begin{cases} 2T(n/2) + O(n) & \text{if } n > K \\ O(1) & \text{otherwise.} \end{cases}$

K is NOT a constant and should appear in your solution

(f) $T(n) = \begin{cases} T(n/K) + O(1) & \text{if } n > K \\ O(1) & \text{otherwise.} \end{cases}$

K is NOT a constant and should appear in your solution

(g) $T(n) = \begin{cases} T(\sqrt{n}) + O(1) & \text{if } n > 1 \\ O(1) & \text{otherwise.} \end{cases}$

(h) $T(n) = \begin{cases} T(2\sqrt{n}) + O(1) & \text{if } n > 1 \\ O(1) & \text{otherwise.} \end{cases}$

(i) $T(n) = \begin{cases} \sqrt{n}T(\sqrt{n}) + O(n) & \text{if } n > 1 \\ O(1) & \text{otherwise.} \end{cases}$

(j) $T(n) = \begin{cases} T(n-1) + O(\log n) & \text{if } n > 1 \\ O(1) & \text{otherwise.} \end{cases}$

2 Summations

(a) Find the asymptotic upper bound on the summation

$$\sum_{k=0}^{\log n} n/2^k$$

(b) Prove that

$$\sum_{k=1}^{n} \frac{1}{k} = \Theta(\ln n)$$

Hint: use integration.

3 Prefix sums

(a) Using induction, rigorously prove the correctness of the non-recursive work-optimal algorithm for prefix sums presented in class.

(b) Write out the pseudocode for the non-recursive work-optimal solution to prefix sums.