CSE 347 Analysis of Algorithms February 2, 2017

Homework 2

Instructor: Brendan Juba/Robert Utterback *Due: February 14, 2017, 11:59 PM*

Reminder: You may work in groups and use outside sources. But, you must write up solutions in your own words and properly reference your sources for each problem. This includes listing your collaborators and properly citing any sources you use. Solutions to each problem must be electronically typeset and submitted online via Blackboard. Instructions appear in the E-Homework Guide: `http://www.cse.wustl.edu/~bjuba/cse347/s17/ehomework/` For all problems in this course, your solutions should provide a proof of *both* correctness and a running time bound, unless the problem explicitly states that this is not necessary.

1. *Kleinberg & Tardos* – Chapter 5, exercise 3
2. *Kleinberg & Tardos* – Chapter 5, exercise 5
3. *Kleinberg & Tardos* – Chapter 5, exercise 7
4. (a) Consider the following problem. You are given an array of integers A of size n, and you wish to compute the *prefix sums* of A. That is, you wish to find an array B such that each $B[i] = \sum_{j=1}^{i} A[j]$. Design and analyze a parallel algorithm for computing prefix sums that uses work $O(n)$ and span $O(\log^k n)$ for some integer k (we call this "poly-logarithmic" span).

   (b) Recall the problem of finding the closest pair of points, discussed in Lecture 5 and Section 5.4 of Kleinberg & Tardos. Show how to parallelize the divide and conquer algorithm we saw in lecture (on pp.230–231 of Kleinberg & Tardos). In particular, describe and analyze a parallel algorithm for finding the closest pair of points in 2-d that still only uses work $O(n \log n)$, but achieves polylogarithmic span. You may assume you have the $O(n)$-work, polylogarithmic-span algorithm for prefix sums that part (a) asks for.