Algorithms in the Real World (15-853), Spring 12
Assignment #4 Due: April 5

Complete all problems.
You are not permitted to look at solutions of previous year assignments. You can work together in groups, but all solutions have to be written up individually. Please submit a hard copy during class.

Problem 1: Fast Unbalanced Quad Trees (10pt)

Describe an algorithm that builds a Quad tree in $O(n \log n)$ time. It has to run in this time even if the quad tree is very imbalanced. Please be concise in your description, ideally using pseudocode. Hint: see description of how to build Callahan-Kosaraju tree in the given time.

Problem 2: Cover Tree Insertion (5pt)

"Algorithm 2: Insert" from the reading on cover trees is written recursively. Give an example of a point set and insertion point in which the algorithm will return multiple levels up the recursion before executing step 3(b).

Problem 3: Closest Pair (10pt)

(a) Write pseudocode that finds the closest pair of a set of points given its tree decomposition generated by the Callahan Kosaraju algorithm (page 43 of the slides). Your code should look something like the code on page 47 for generating the realization. Note that you don't need to generate the realization, but are just asked to return the single closest pair.

(b) The slides outline how to generate the nearest neighbor for all points instead of just the overall closest pair. As mentioned in the slides, however, step 2 (page 52) might not be efficient without some modification. Give an example where it would not be efficient.

Problem 4: Parallel Maximum Contiguous Subsequence Sum (10pt)

The *maximum contiguous subsequence sum* (MCSS) problem is: given a sequence of numbers (positive and negative), return the sum of the contiguous subsequence that is largest. For example, for

$$[2, -3, 2, -1, 4, -2, 3, -4, 2, -3, 4]$$

the answer is 6 $(2-1+4-2+3)$. For each of the following two solutions you can use pseudocode.

(a) Describe a linear work, $O(\log n)$ span parallel divide-and-conquer algorithm for the MCSS problem.

(b) Describe a linear work, $O(\log n)$ span parallel algorithm for the MCSS problem that only uses unnested parallel loops and a constant number of calls to reduce and scan.