Parallel Algorithms (15-492), Fall 07
Assignment #4 **Due: October 30th**

Instructions. This is a short assignment with only 1 problem. It will be due at the beginning of class on Tuesday October 30, 2007.

Problem 1: Finding the Closest Pair in 3-d

Given a finite set of points $X \subseteq \mathbb{R}^d$ and a Euclidean distance metric $d(\cdot, \cdot)$, the problem of finding the closest pair is to report a pair of points that are closest together. As seen in lectures, on n two-dimensional points, a simple divide-and-conquer algorithm can solve this problem in work $O(n \log n)$ and depth $O(\log^2 n)$. You may wonder what can be said about higher dimensions.

It is known that for n d-dimensional points, an extension of the divide-and-conquer algorithm can solve the closest-pair problem in work $O(n \log^{d-1} n)$ and depth $O(\log^d n)$. In this problem, you will explore the 3-d case: design and analyze an algorithm for the 3-d closest-pair problem with work $O(n \log^2 n)$ and depth $O(\log^3 n)$.

(**Hint:** If you use the same outline as the 2-d version, when you get to finding the closest pair within the 2δ "slap," try to reduce it to the 2-d case. It will not be exactly the 2-d problem we saw in class, but it should be pretty similar.)