**Design Idea Worksheet – Print out and fill out one of these for each design idea.**

**Your name: Joshua Ruebusch**

**Design # (each of your designs should have a number):1**

**This is a (circle one): --New design**

**--Variation, based on design # \_\_\_\_\_\_\_\_ (If it is a variation, you only need**

**to fill in the parts that differ from the basic design)**

**Data structure on which design is based:**

A binary search tree implemented with a linked list

**Describe how the data will be organized:**

Data will be sorted with a K-d tree.

**Augmentations**: None of the data structures we have studied fit this problem perfectly, all of them have performance problems if used as-is. In this space describe any changes or additions you are making to the basic data structure to improve performance.

Typically a search tree does not have alternating layers of x coordinates and y coordinates. This one will in order to consider the type of data that it holds. This allows for a way to somewhat sort two dimensional data.

**Performance**: For each of the main methods of the Starbucks.h, describe in detail the algorithm you would use, and then give the asymptotic running time that would result.

**build**

**Algorithm:** For this algorithm I want to read in each coordinate one at a time from the file and put it into an array. Then shuffle the array as this will give a better chance of getting a midpoint for the starting of the dividing of the data. Once shuffled go through the data in the array and use a recursive “insert” method to create a K-d tree that alternates x and y coordinates with each layer. (Note: base case is when point reached is NULL and is inserted there otherwise method traverses left or right, dividing the tree) The way this is set up, each division will go through exactly one starbucks location.

**Running time:** Since this method involves cycling though all of the data the asymptotic running time is . Though the method is recursive each point of the array is traversed once and creates one node on the tree. Cannot explicitly prove this, but seem as though this would be the case.

**getNearest**

**Algorithm:** This method will take in three parameters, x coordinate, y coordinate, and a Boolean to keep track of which level the tree is on (X or Y). The method will recursively call itself to move down the tree using the coordinates that were inputted to make decisions on how to traverse the tree. Since, as is, this will not take into consideration that the closest starbucks might be right on the other side of the first division, this method will also keep track of how far the found starbucks if from the initial division. If this length is larger than the starting distance to the selected starbucks it will travers the other side of the tree. Otherwise the correct starbucks has been found and will return the location of it.

**Running time:** Depending on how the method has to run, this could be two different answers. If the method only has to travers the first half of the tree and passes the distance from initial division test then the runtime will be . However if the other side of the tree has to be traversed then it will have gone through all of the nodes making the runtime

**Accuracy**: How accurate do you expect your solution to be? 1.0 is perfect, 2.0 means off by a factor of 2, and so on. Explain your reasoning.

I expect this solution to have an accuracy of 1.0 because it will search for the closest location as it goes through each division of the data. The only way that it would lose accuracy is if I didn’t have a way to check the other side of the first division if there were a doubt that the closest one might be on that side.

**Difficulty:** On a scale of 1 to 10, with 1 being “very easy” to 10 being “impossible,” how hard will it be for you to successfully implement this solution in the time allotted (1 week)?

Probably about a 7. This is definitely not the easiest solution to the problem, but it is one of the more efficient. It doesn’t seem so excessively difficult though that it would be impossible to make in a week.