1. buildFoodCodes
   1. read spreadsheet Food-STDV.xls into an array of the food codes of each item
2. getList
   1. Read parcel identifiers into array
3. getList
   1. Read storelist into an array
4. buildFoodCostList
   1. Using FINAL-Prices.xls
   2. Foreach store and each item of food read the price into a matrix
5. Call parallel computing
6. Jobserver.submit
   1. Make calls to parallel computing
7. Foreach parcel
   1. buildTravelCost
      1. Searching for the closest 15 stores can change
      2. From each parcel to every store and each store to store calculate the cost of traveling
   2. buildCostList
      1. From every item in every store calculate the travel cost plus the retail cost of the food
         1. For example to travel from Apples to Bananas in the same store is the retail cost of that movement. But going from Apples in one store to Bananas in another store is retail cost of the item *plus* the cost of transportation from the one store to the next. (This is the key heuristic)
   3. minSpanningRec
      1. build a minimum spanning tree based on all of the items in all of the stores.
   4. finishTree
      1. Based on the stores visited, find all of the cheapest items in the stores of the min spanning tree
   5. calcSet
      1. Calculate the set of all items that were purchased and the total cost of transporation
   6. Return the near mincost and the final list of items purchased
8. Store results in finalresults.csv