Brian Metz

For the group project I did both of the shortest distance algorithms. The first one, and the more complicated of the two gets passed a vector of float vectors, and an int vector. The nested vector which will now be called v contains all the distances that connect the wineries. The outer vector represents an individual winery, while the inner vector at that position has it’s unique number in the 0 position followed by all the distances connecting it to the other wineries. The int vector that gets passed in contains in numerical order which wineries the user wants to visit. The first half of the algorithm, which runs in O(n3) compares the winery number of the outer vector to the wanted values of the int vector. If it’s a match it goes into the inner vector and iterates though the entire list popping off values that aren’t wanted. Once the list is trimmed down the algorithm looks for the shortest distance from the villa, it then goes to that outer vector, pops the vector of villa distances, and iterates though all the inner vectors popping off the value of the winery we just visited. This continues until the vector is empty and returns the total distance as a double.

The second algorithm runs similarly. It gets passed a vector of double vectors, but this time gets passed a blank int vector, and two ints. The blank int vector gets populated within the function for the order in which these wineries will be visited. The first of the ints represents the starting winery, and the second int is the total number of wineries to visit. While not identical, the processing for both the algorithms runs in O(n3) and the second shouldn’t require much explanation.