FOOD DELIVERY

For the first test group m<=100, k=3, we can check all triples of pairs and if they form a "triangle" then thats a possible route to choose. The solution may be found in sol1.cpp.

For the second test group m<=1000, k=3, instead of checking all the possibilities we can do the following: suppose stores i and j are connected and stores i and k are connected, now all we have to do is see if j and k are connected. If we have all the roads sorted, we can do binary search or even lookup table to do it faster. In sol2.cpp we have used binary search.

For the next test group m <= 50, k <= 5, we can use some kind of brute force (maybe with little optimization, using some tricks with sorting, binary search etc.), but we are not going in further details about this solution.

Let’s take a look at the whole picture. We are given undirected graph and we have to count all the cycles of length k int the graph. A cycle of length k means that the cycle contains k vertices and k edges. To solve this problem Depth First Search (DFS) can be used. Using DFS we can find all the possible paths with length k-1 for a given starting point. Notice that we checked for path of length k-1 because the n-th edge will be the closing edge of cycle. Now all we have to do is check if the path ends with the vertex we started from, if yes then we count this as cycle of length k. This solution may be found in sol3.cpp.