We sort d_i into an increasing sequence by merge sort $O(n\log n)$. Use a binary search on i to find the smallest d_i which meets the requirements $O(n\log n)$. Specifically, we fix i and consider only roads which take d_i driving hours or less to travel from a warehouse to the corresponding shop. We construct a corresponding bipartite graph with the warehouses as vertices on left side and with shops as vertices on right side. For each road with driving hour $d_j \leq d_i$, construct a corresponding edge with weight d_j between the warehouse vertex and shop vertex. Form a super source for warehouses with edges of infinite weight and a super sink for shops with edges of infinite weight. We now run the Edmons-Karp algorithm to find the maximal flow through such a network to see if they are enough to obtain a matching of warehouses with shops which is of size n $O(|n||m|^2)$. If we can find such a matching, which means it is possible to supply all shops only considering roads that take d_i driving hours or less, we use binary search to find a smaller i. Otherwise, we use binary search to find a larger i. Then test i with constructing another graph and applying Edmons-Karp algorithm again. Repeat finding and testing i, until we find minimum i that meets the requirements. The matching of i's corresponding graph is the roads to send goods from warehouses to shops.

Time complexity: Sorting an array + using binary search to find minimum i + constructing a network and running the Edmons-Karp algorithm in binary search $= O(n\log n) + O(n\log n) * O(|n||m|^2) = O(n^2m^2\log n)$