We sort into an increasing sequence by merge sort . Use a binary search on to find the smallest which meets the requirements . Specifically, we fix and consider only roads which take 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 , construct a corresponding edge with weight 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 . If we can find such a matching, which means it is possible to supply all shops only considering roads that take driving hours or less, we use binary search to find a smaller . Otherwise, we use binary search to find a larger . Then test with constructing another graph and applying Edmons-Karp algorithm again. Repeat finding and testing , until we find minimum that meets the requirements. The matching of ’s corresponding graph is the roads to send goods from warehouses to shops.

Time complexity: Sorting an array + using binary search to find minimum + constructing a network and running the Edmons-Karp algorithm in binary search