24-3-2.

The shortest distance from A -> D is 2, via A -> C -> B -> D. Dijkstra’s algorithm gives a shortest distance of 4 (A -> B -> D).

24-3. a.) \( R[i_1, i_2] \cdot R[i_2, i_3] \cdots R[i_{k-1}, i_1] > 1 \),

\( (1/R[i_1, i_2]) \cdot (1/R[i_2, i_3]) \cdots (1/R[i_{k-1}, i_1]) < 1 \) on taking reciprocal

\( \log (1/R[i_1, i_2]) \cdot \log (1/R[i_2, i_3]) \cdots \log (1/R[i_{k-1}, i_1]) < 0 \) on taking log

Now, create a directed graph with \( n \) vertices, each of which represents each currency. Each vertex then has \( n - 1 \) outgoing edges to other \( n - 1 \) vertices with the weight \( \log (1/R[i, j]) \). Add one additional vertex as a source with \( n \) outgoing edges to \( n \) vertices, respectively, with the weight 0. Then, the original problem is converted to the problem of finding a negative-weight cycle in a graph. Using Bellman-Ford algorithm we can detect such a negative-weight cycle can do it.

Since \( O(V) = O(n) \) and \( O(E) = O(n^2) \), the total running time is \( O(n^3) \).

b.) First, run the Bellman-Ford algorithm as described above. Then, if it says there is a negative-weight cycle, just relax all the edges once more. It will be seen that the \( d \) values of some vertices change. The sequence found by this approach can be printed by using the \( \pi \) values. The total running time is \( O(n^3) \).