"Supermarket Optimization" Problem

You've been hired as a market consultant to try and help a local supermarket come up with better placement of items based on buyer's preferences, and towards that goal you'd like to identify items that are frequently purchased together based on existing records of buyer's transactions.

You are given as input:

- A transaction database—a file consisting of a single row per transaction, with individual product's SKUs given as space-separated integers. A single transaction consisting of products with SKUs 1001, 1002 and 1003 would have a line that looks like: '1001 1002 1003'
- A minimal 'support level' parameter, *sigma* – a positive integer.

Implement, in Python, an efficient algorithm for generating all *frequent item sets* of *size* 3 or more; groups of 3 or more items that appear together in the transactions log at least as often as the support level parameter (sigma) value.

For example, given a value of sigma = 2, all sets of 3 items or more that appear 2 or more times together in the transaction log should be returned. The results should be returned as a file with the following format: <item *set size (N)>, <co-occurrence frequency>, <item 1 id >, <item 2 id>, ... <item N id>.* **Run the algorithm on the attached transaction log file and provide the results obtained for a value of sigma = 4.**

If you have any questions don't hesitate to get in touch for clarification. We'd like the deliverable to be in Python as mentioned above, and hosted on GitHub in a private repository with a README clearly explaining how to run the code. Extra points for coding style and comments throughout explaining non-obvious sections in the code.

A transaction database file to download:
https://storage.googleapis.com/assets.beehive.ai/data/retail_25k.dat