CENG4120 Report

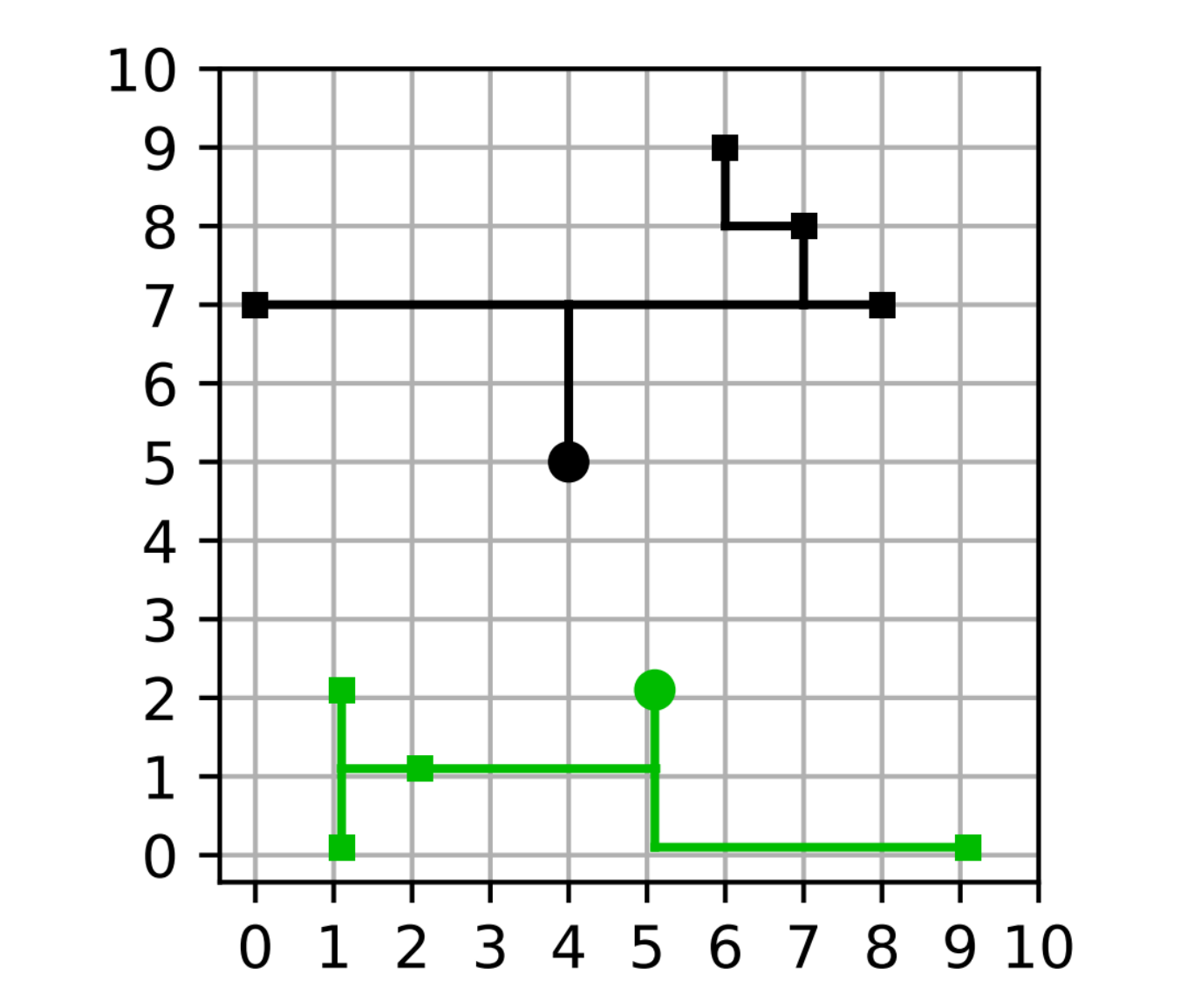
Report on Clock Tree Synthesis

Group Team 08

Wong Wai Chun 1155173231

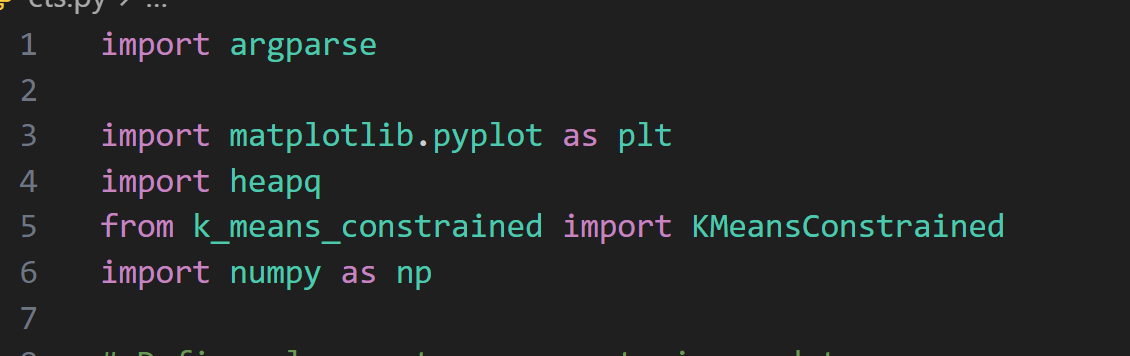
Yip Ching Wai 1155145400

**Overview**



This Python script provides a comprehensive solution for Clock Tree Synthesis (CTS), which is a crucial step in the design of circuits to ensure that the clock signal reaches all elements of the system in a synchronized fashion. It reads configuration parameters and node information from an input file, processes this data through different routing algorithms, and outputs the results to a specified file.

**Script Structure and Modules**



Argparse Module: Used for parsing command-line options, enabling the script to accept inputs and outputs from the user.

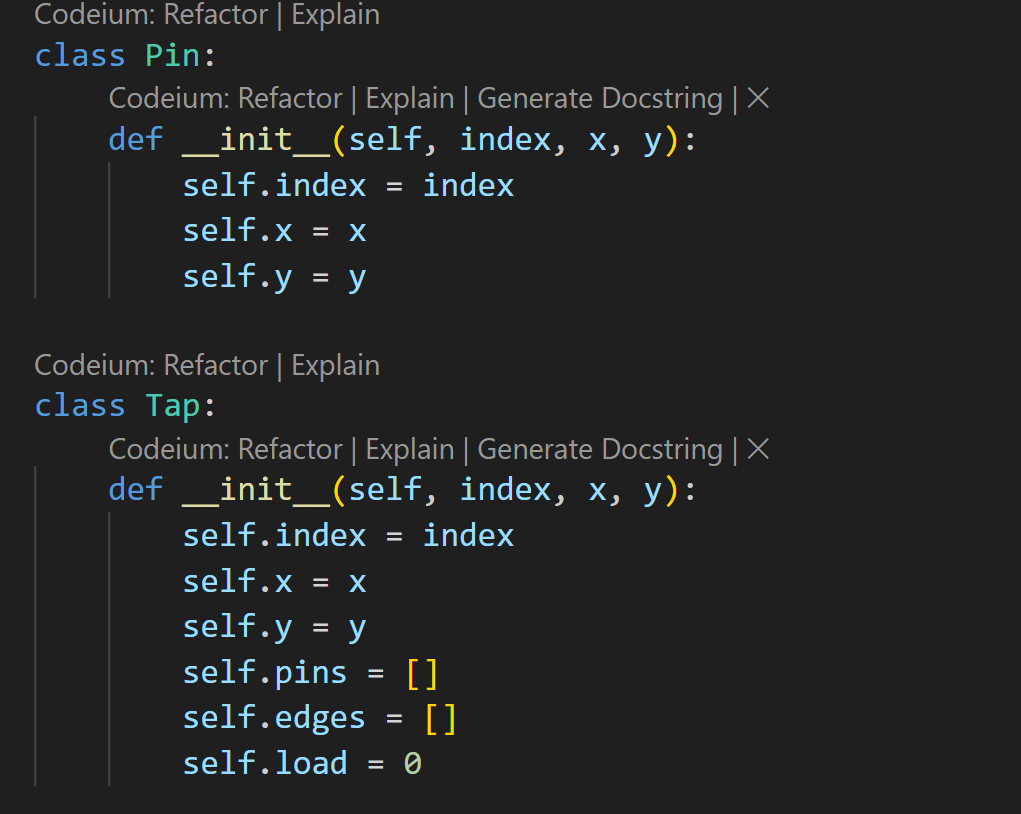
Matplotlib and Numpy: Utilized for plotting the distribution and connections of pins and taps on a grid and for handling numerical operations and optimizations, respectively.

Heapq: Facilitates efficient priority queue management, which is vital for some of the routing algorithms used.

KMeansConstrained: An implementation of the k-means algorithm that adheres to specific constraints, ideal for clustering pins to taps based on specified rules.

**Main Components**

* Data Structures:



Pin and Tap classes model the essential components of the CTS problem.

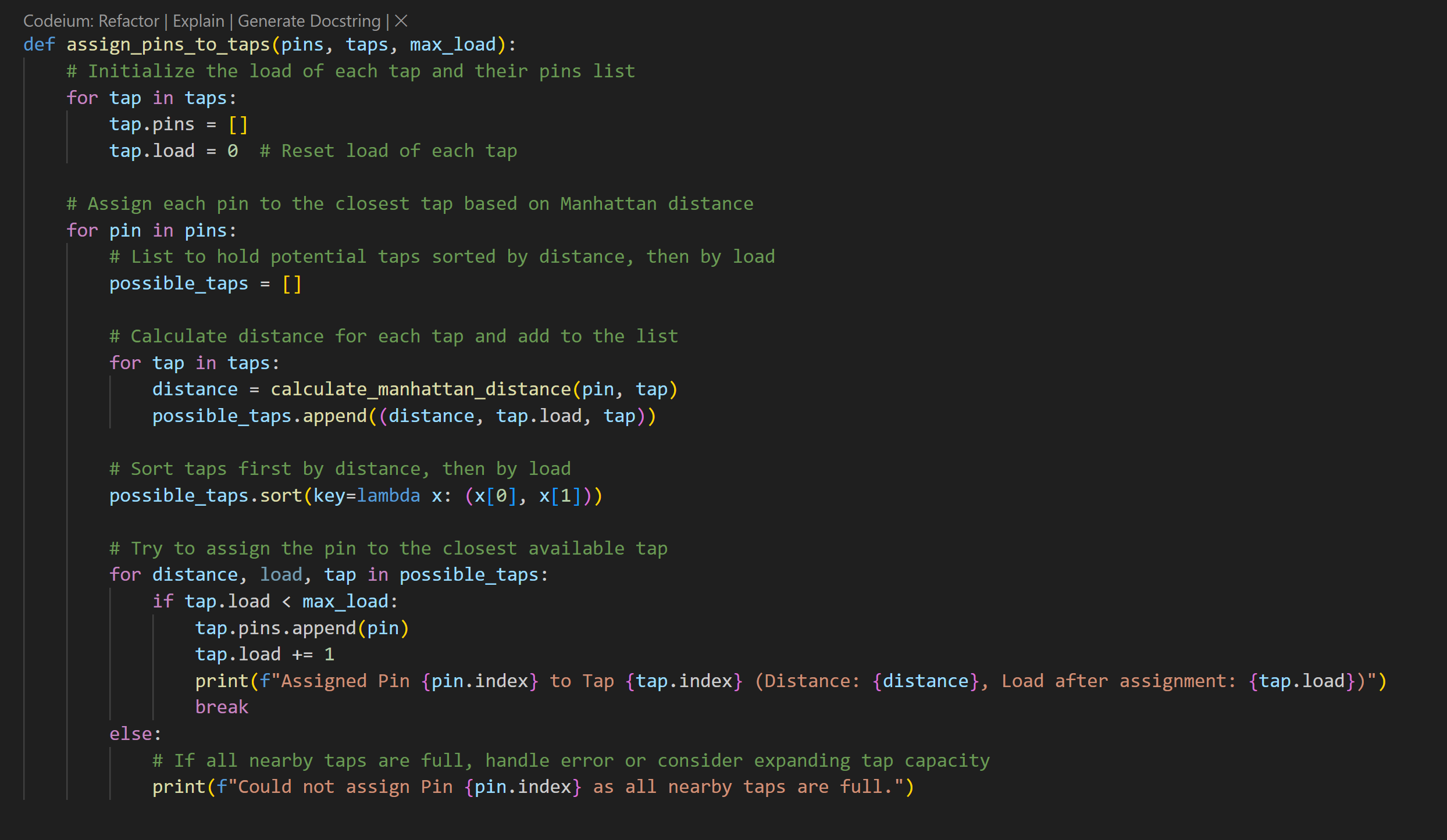
Node and Tree classes are used in hierarchical routing to manage the connectivity of pins and taps.

* Input Parsing (parse\_input):

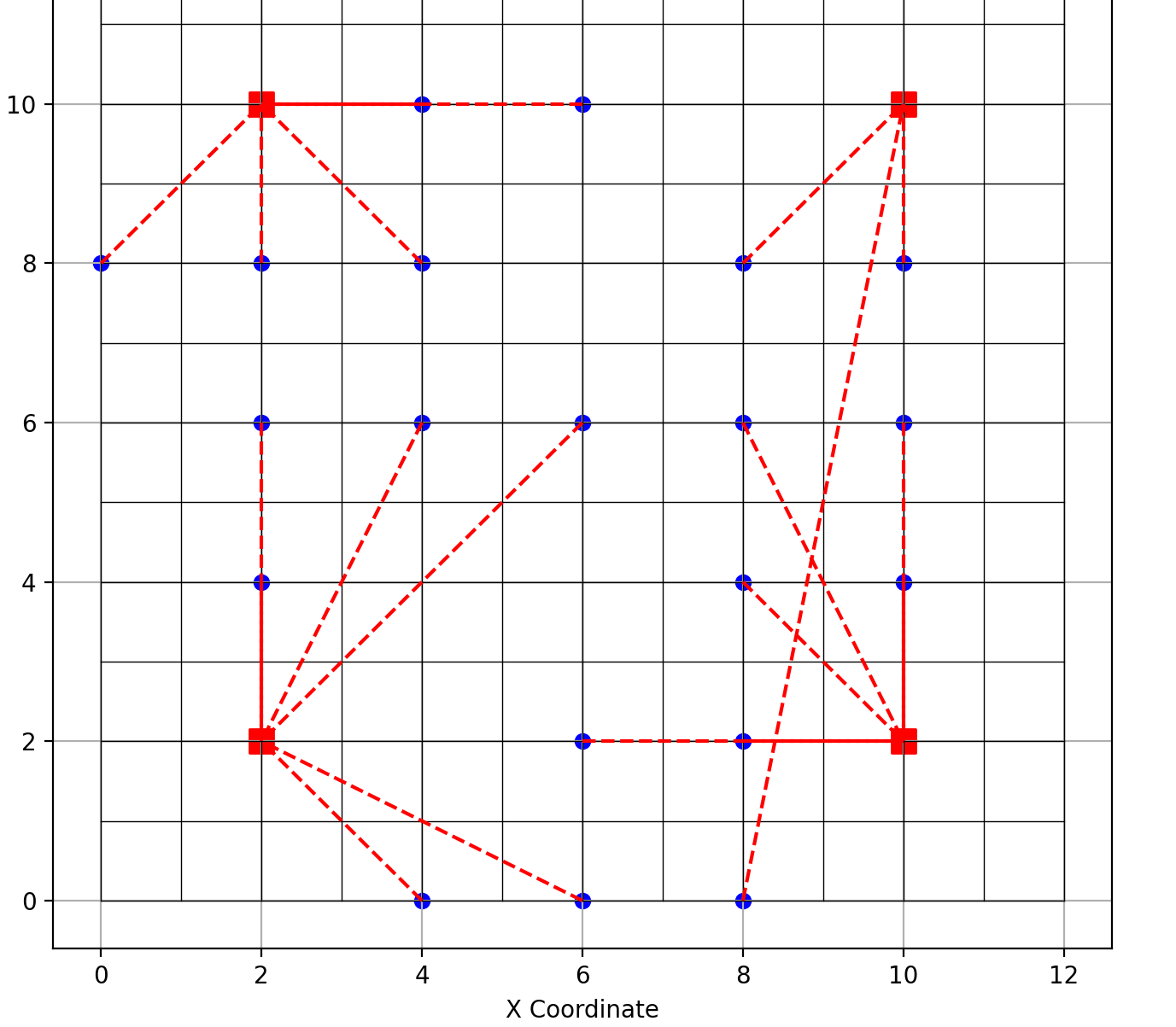
Reads and interprets an input file specifying maximum runtime, load constraints, grid size, and the coordinates of pins and taps.

* Grouping methods

**Direct Assignment** (assign\_pins\_to\_taps): Assigns pins to the nearest available tap based on Manhattan distance, adhering to load constraints.

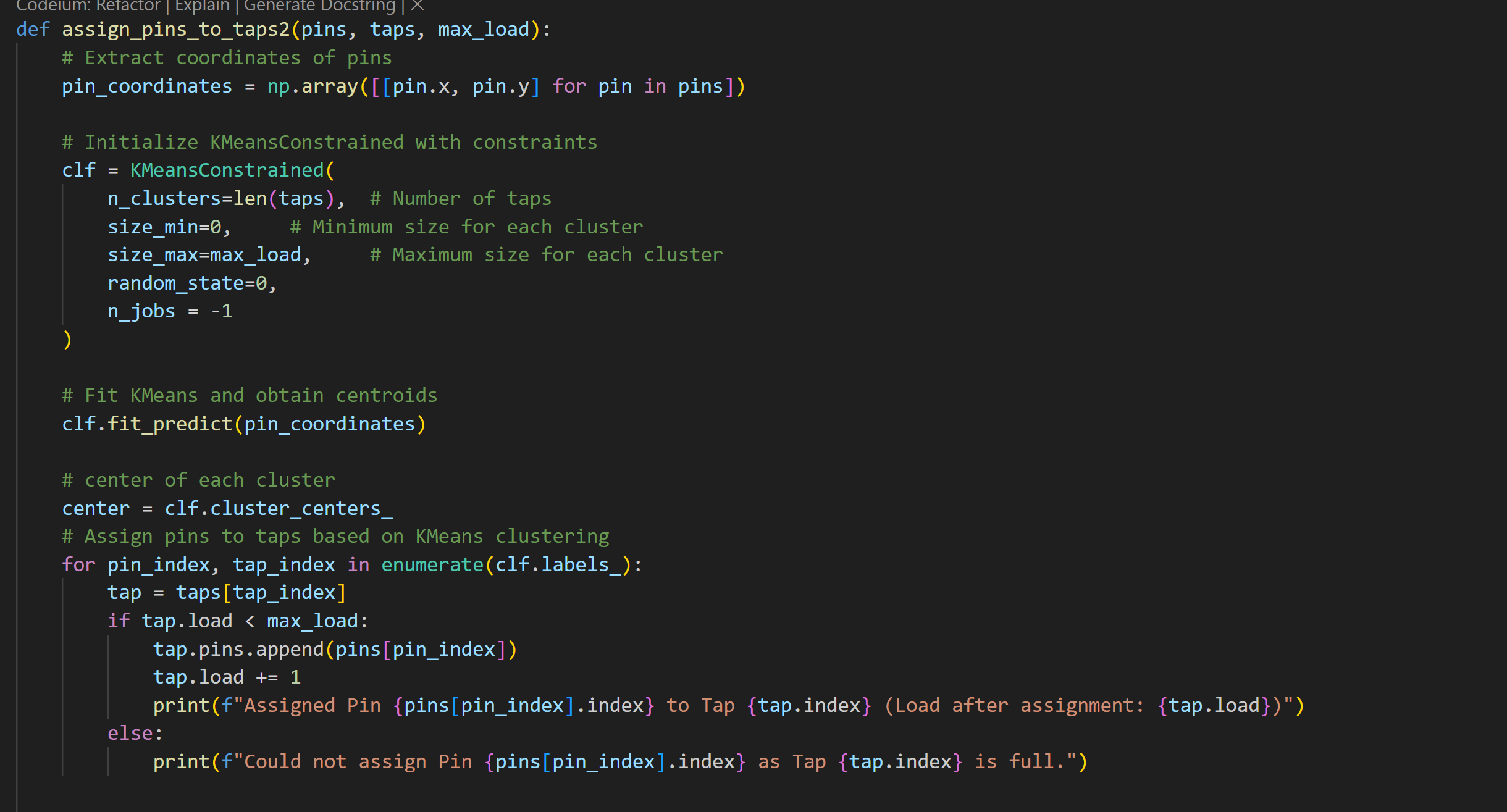


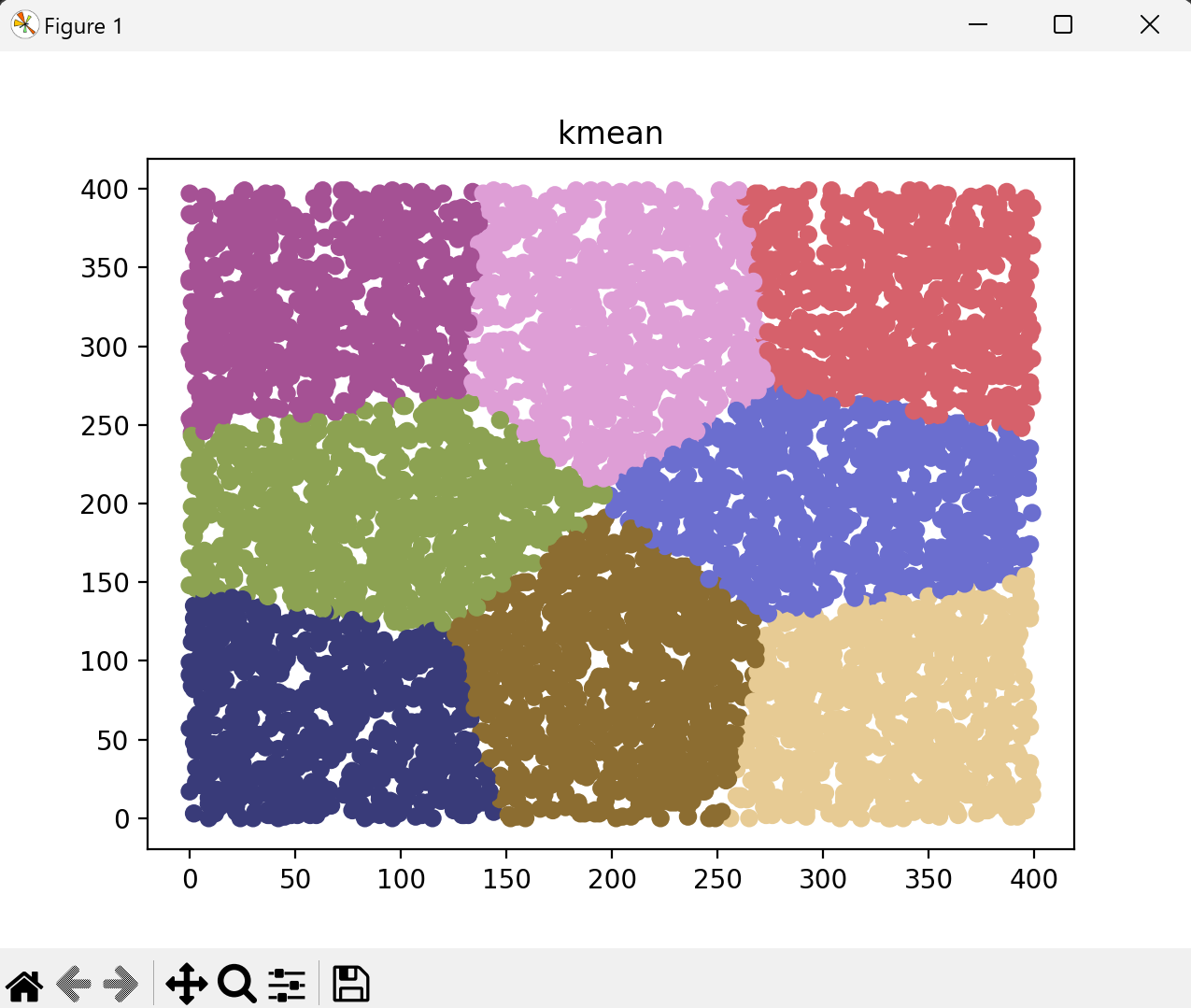
Problem of Direct Assignment



Since it directly assigns the pin to tap has a huge problem when the tap max load is full, the pin can be assigned far aways from the closer pin and cause unexpected results. So we decided to use a better method to group all the pin and tap.

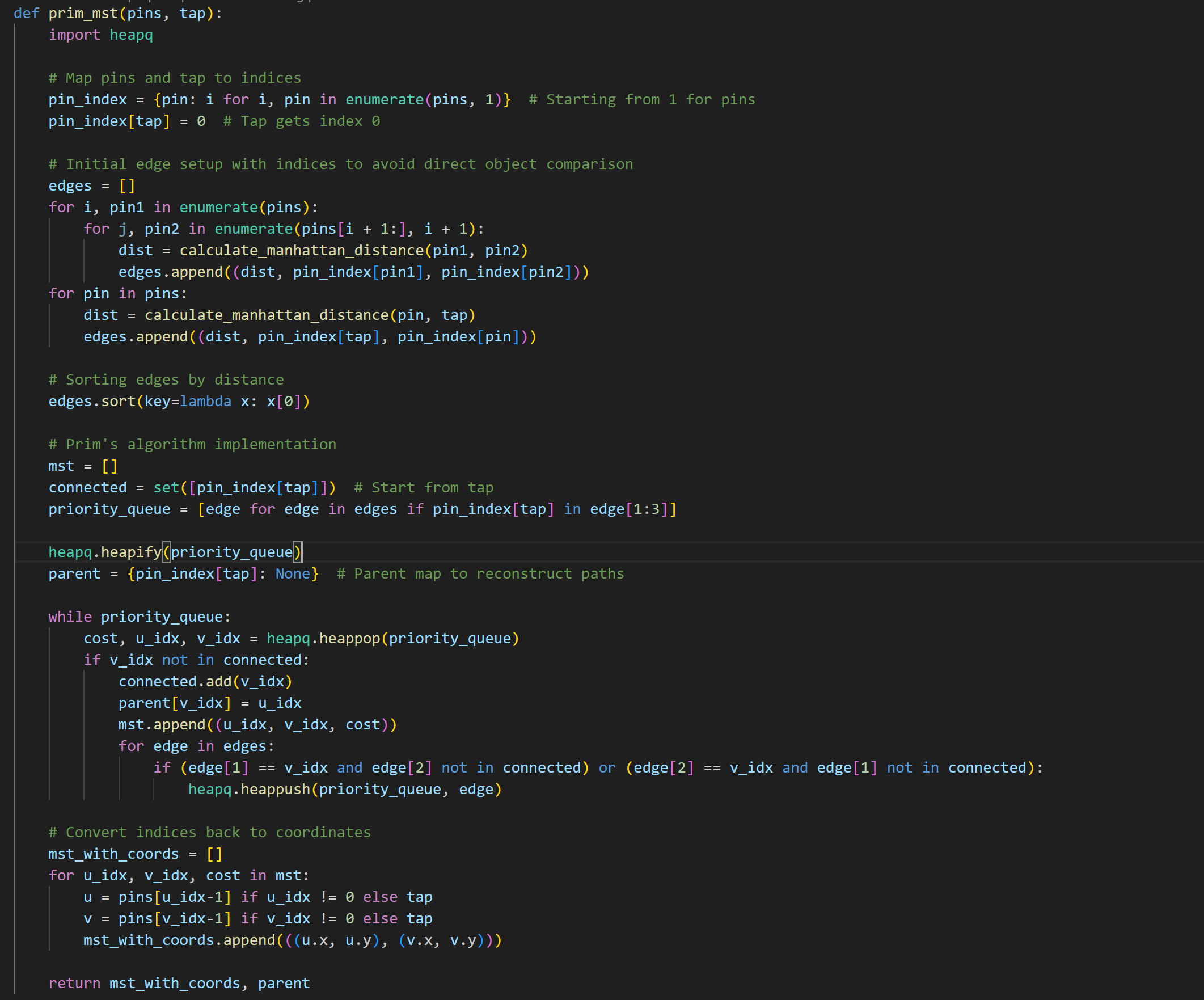
**Constrained Clustering** (assign\_pins\_to\_taps2): Utilizes a constrained k-means clustering algorithm to distribute pins optimally among taps, respecting maximum load constraints per tap.



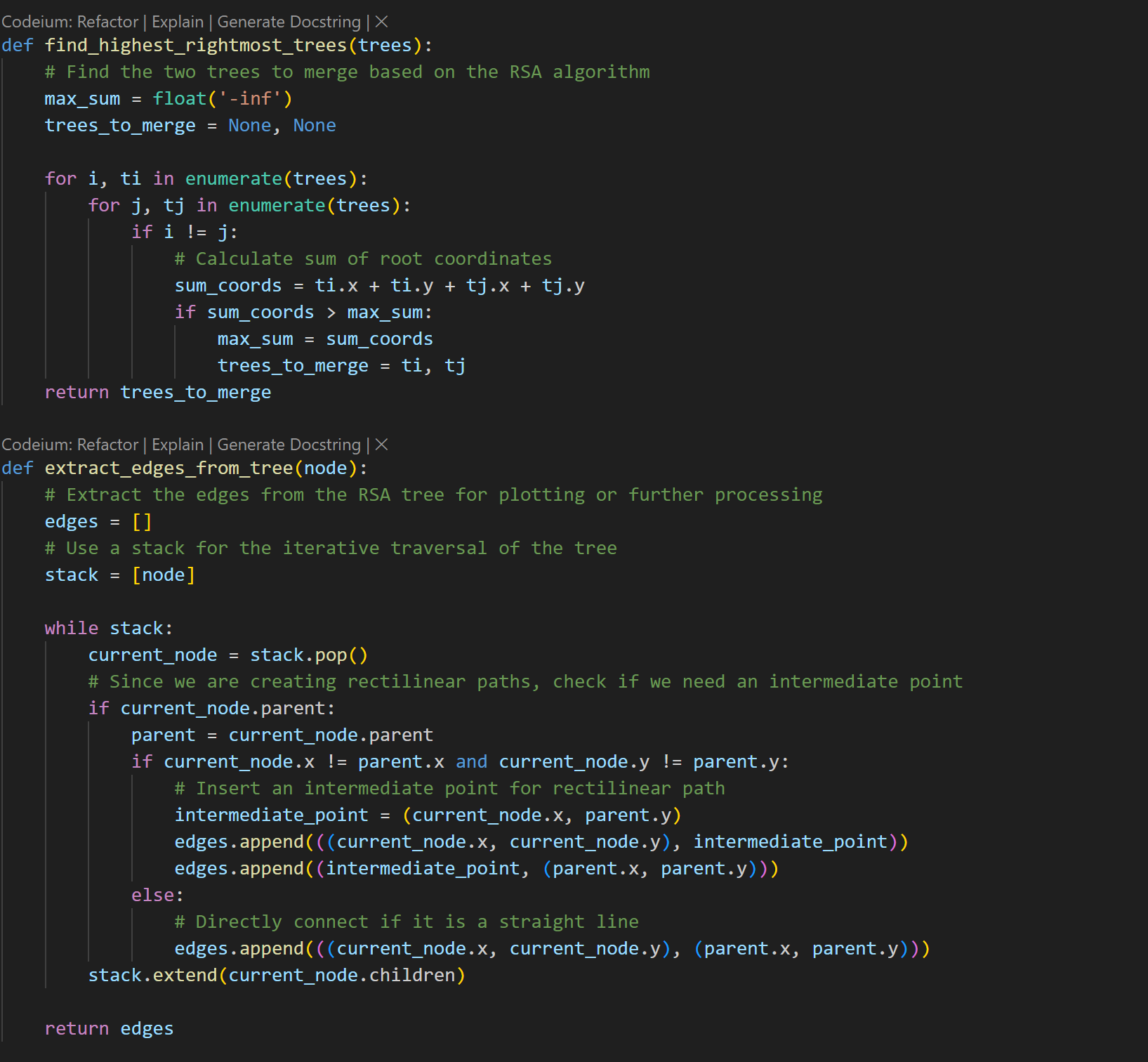
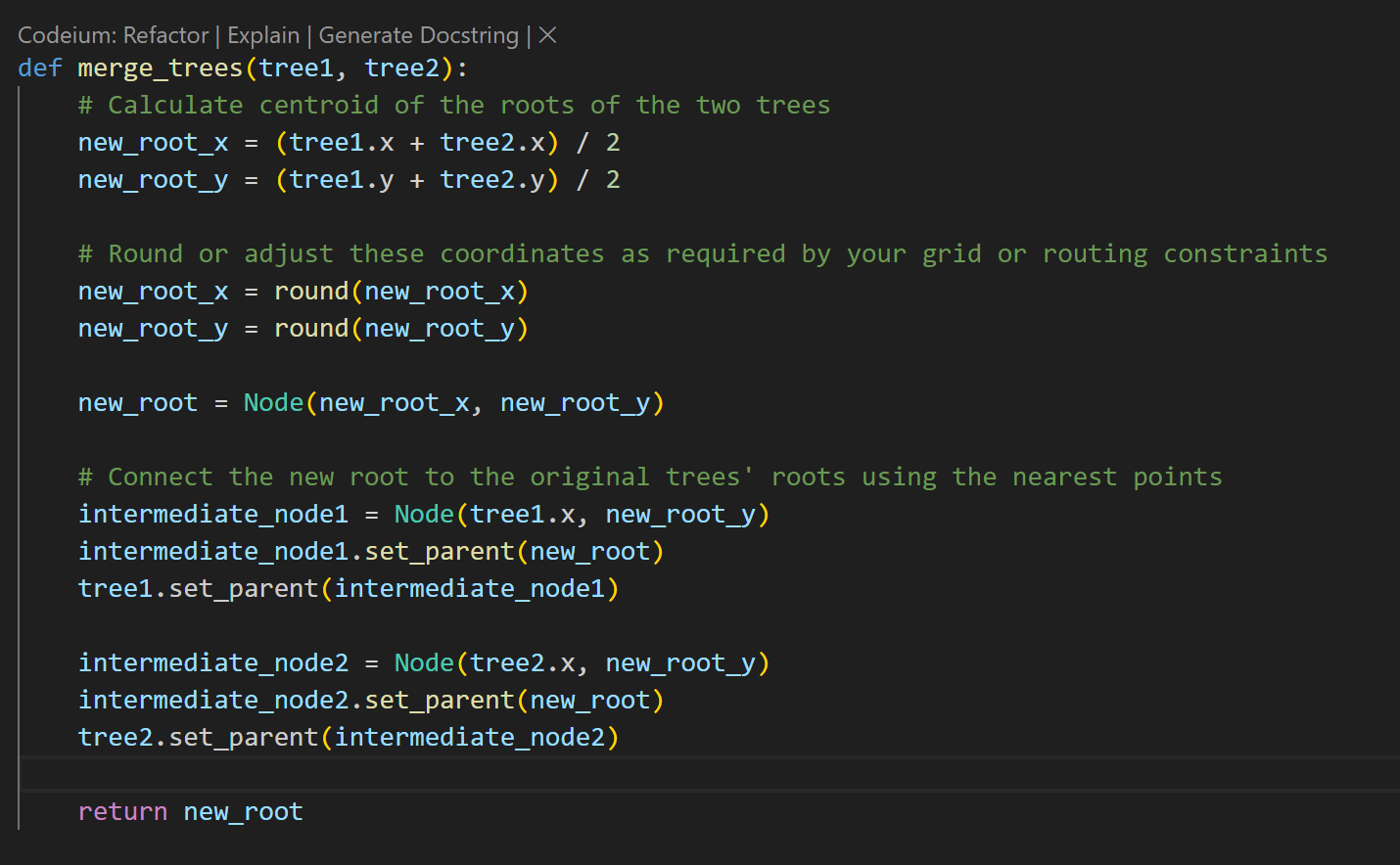
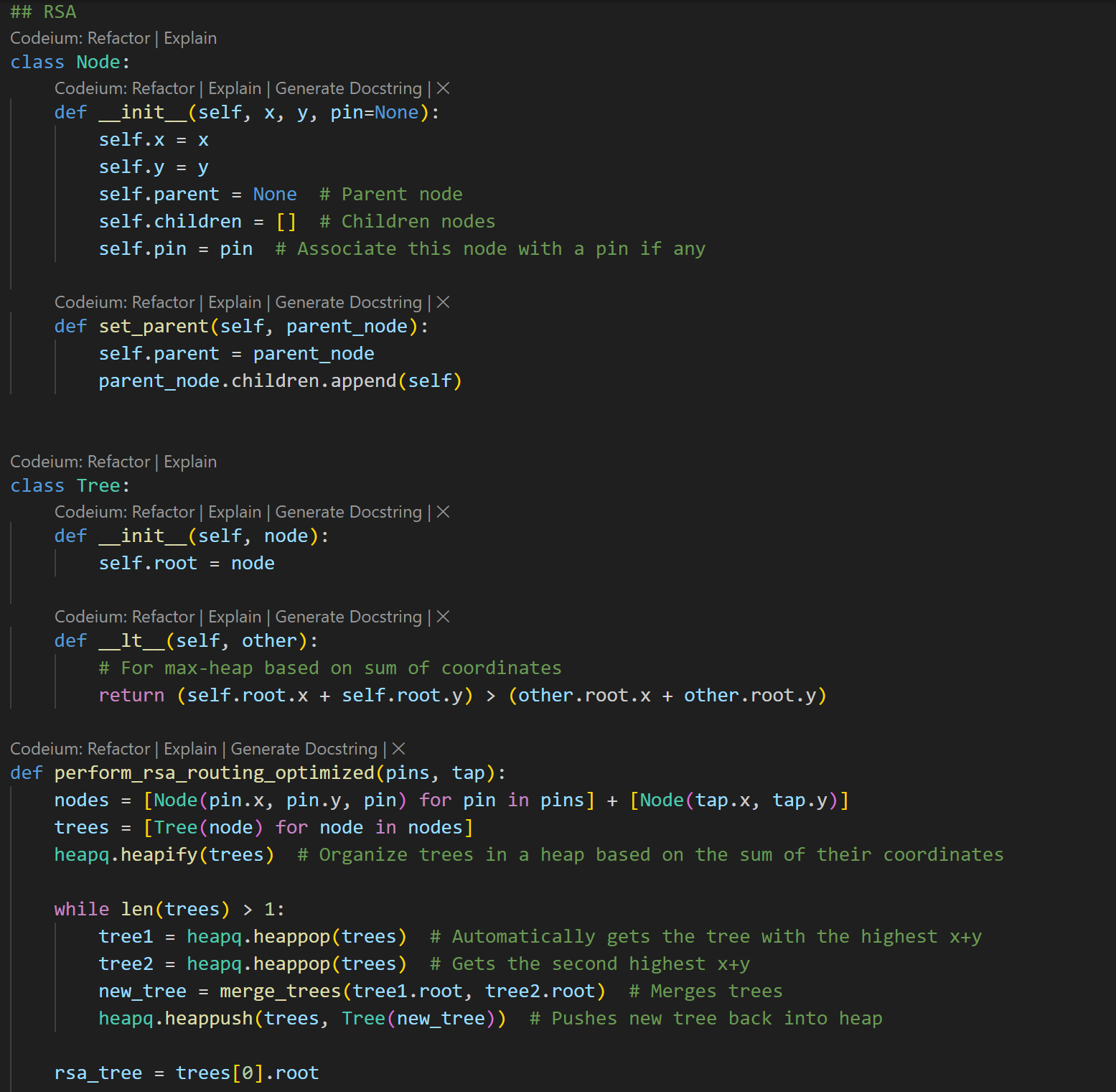
Result:

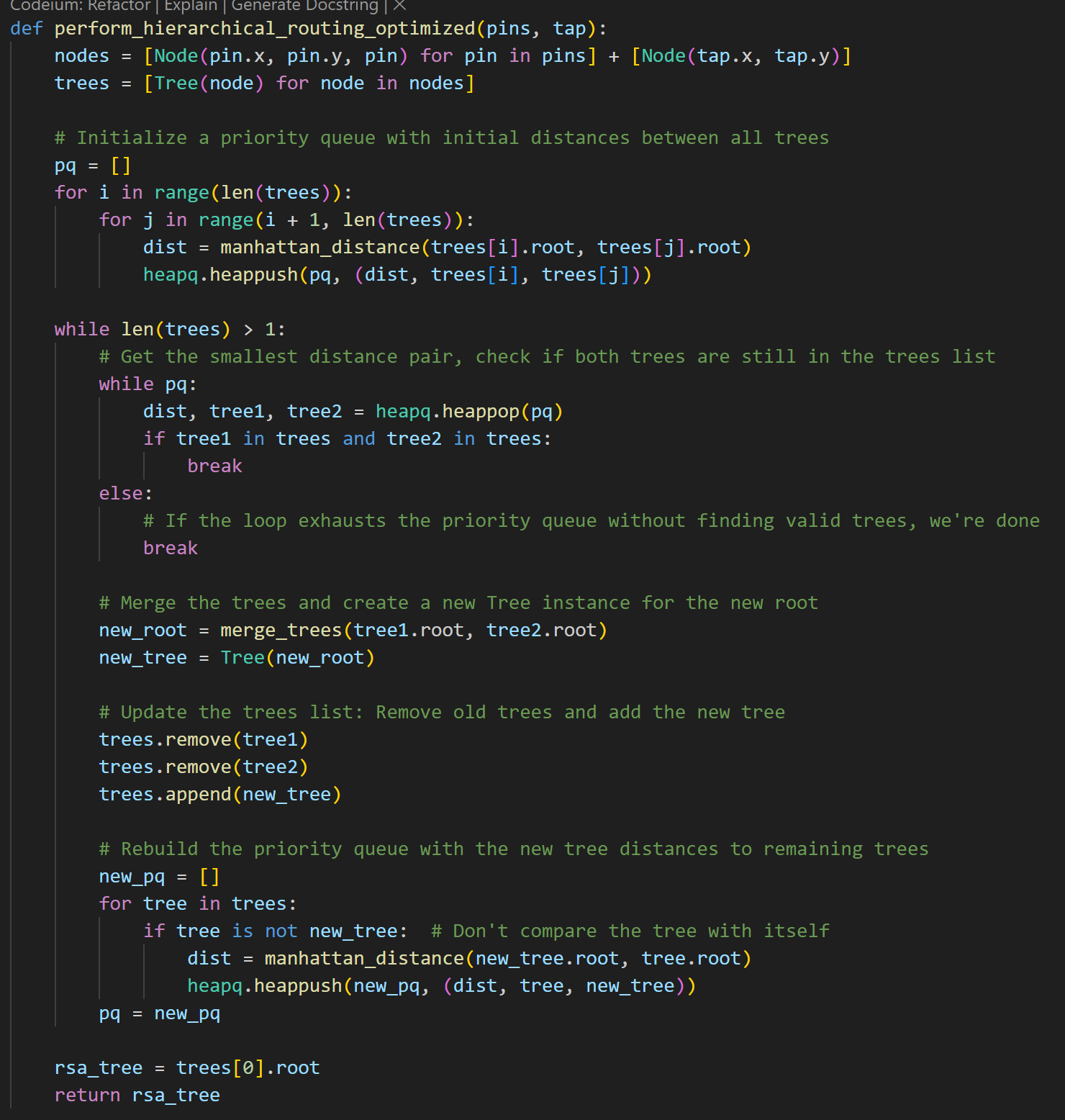
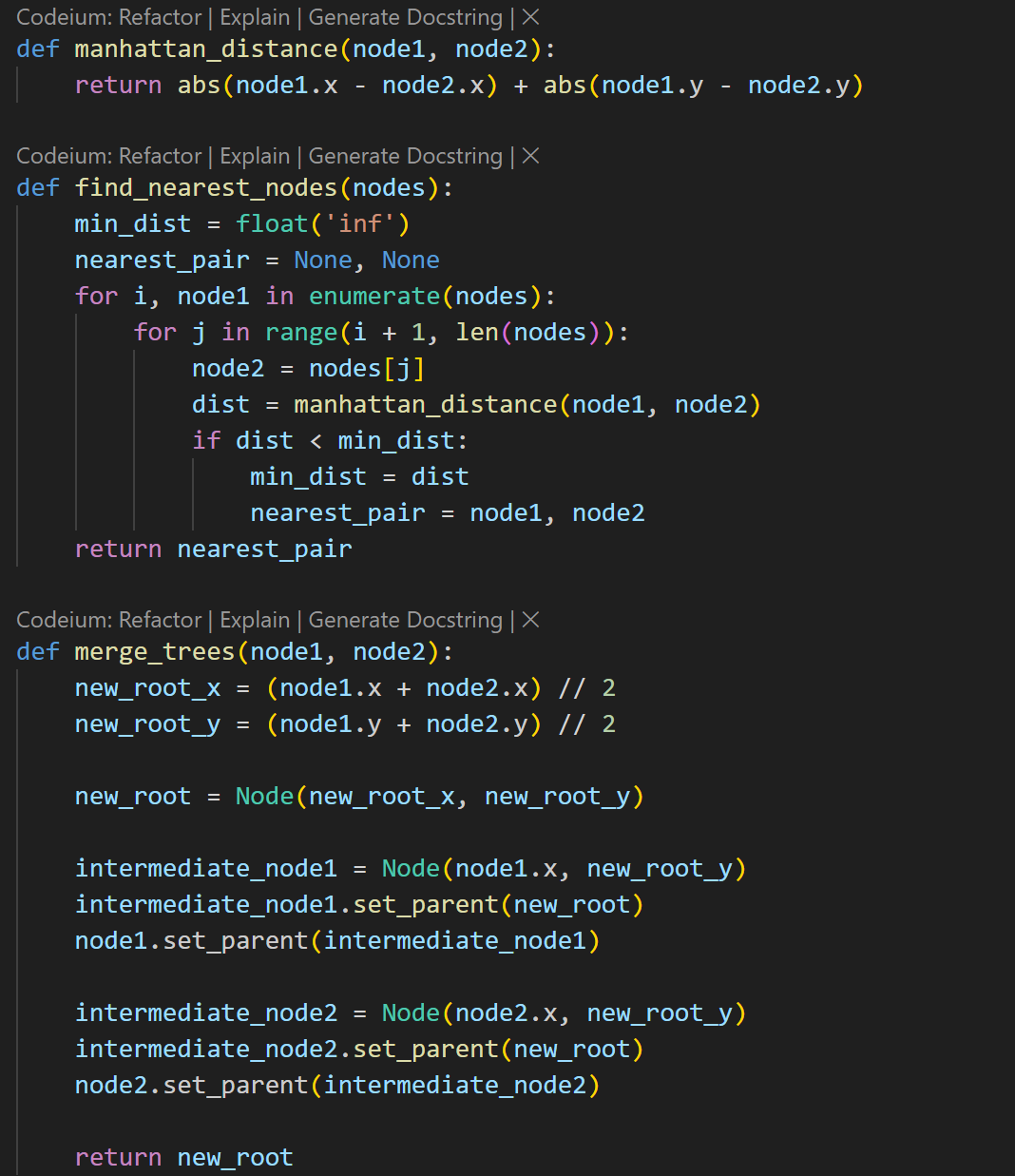
* Routing Algorithms:

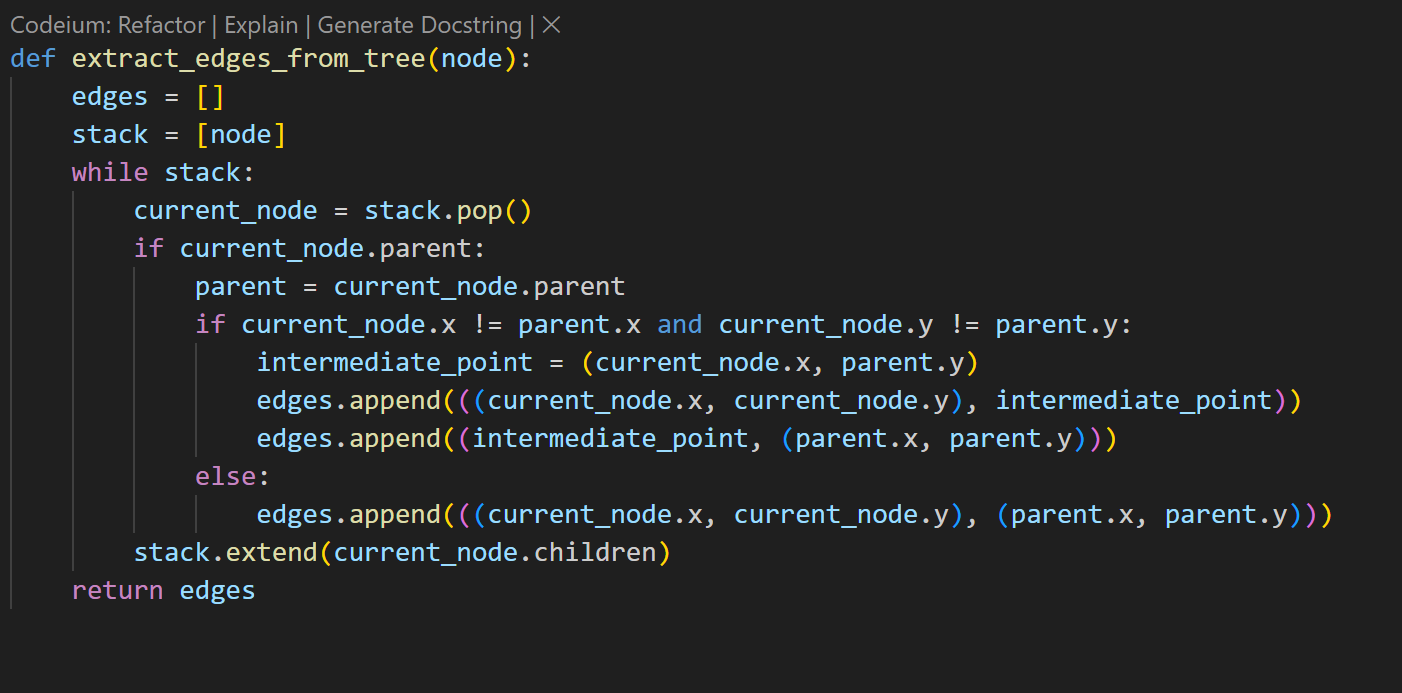
Prim’s Algorithm for MST (prim\_mst): Constructs a minimum spanning tree to minimize the total wire length connecting pins and taps.



Recursive Spatial Algorithm (perform\_rsa\_routing\_optimized): Uses a heap-based approach to merge nodes recursively until one tree remains.



Hierarchical Routing (perform\_hierarchical\_routing\_optimized): Another approach where nodes are merged based on their proximity to optimize the routing tree structur



Output and Visualization:

Plotting (plot\_assigned\_pins\_taps and plot\_connections): Visual representations of the grid and connections for analysis and verification.

Output Writing (write\_output\_file): Documents the final connections in a specified output file format.