class Tag:

def \_\_init\_\_(self**,** name**,** description=None):

self.name = name

self.description = description

#For Printing

def \_\_repr\_\_(self):

return f"Tag(name={self.name}, description={self.description})"

class TagGraph:

def \_\_init\_\_(self):

self.tags = {}

self.graph = {}

#Add tag to the graph

def add\_tag(self**,** tag):

if tag.name not in self.tags:

self.tags[tag.name] = tag

self.graph[tag.name] = {}

else:

print(f"Tag '{tag.name}' already exists.")

#Remove tag from the graph

def remove\_tag(self**,** tag\_name):

if tag\_name not in self.tags:

print(f"Tag '{tag\_name}' does not exist.")

return

for connected\_tag in list(self.graph[tag\_name].keys()):

del self.graph[connected\_tag][tag\_name]

del self.graph[tag\_name]

del self.tags[tag\_name]

def link\_similar\_tags(self):

# Dummy function: currently does nothing

pass

#For printing

def \_\_repr\_\_(self):

return f"TagGraph(tags={list(self.tags.keys())}, graph={self.graph})"