# Code Implementation

import time

import numpy as np

class Bankers:

"""implement Bankers algorithm"""

def \_\_init\_\_(

self, curr\_alloc: np.ndarray, max\_ask: np.ndarray, available: np.array

):

self.allocation = curr\_alloc

self.max = max\_ask

self.available = available

self.need = max\_ask - curr\_alloc

self.num\_processes = len(curr\_alloc)

self.num\_resources = len(available)

self.running = np.ones(self.num\_processes)

def is\_safe(self) -> None:

print()

print("--------------------")

print()

sequence = []

while np.any(self.running):

allocated = False

for p in range(self.num\_processes):

if self.running[p]:

if all(

i >= 0

for i in self.available - (self.max[p] - self.allocation[p])

):

allocated = True

print(f"Process {p} is running")

time.sleep(0.5)

sequence.append(p)

self.running[p] = 0

self.available += self.allocation[p]

if not allocated:

print("Unsafe - no possible sequences")

return

print()

print(f"Safe sequence found!\nSafe sequence: ", end="")

for p in sequence:

print(f"{p} ", end="")

print("\n")

print("--------------------")

print()

return

def main():

Allocation = np.array([[0, 1, 0], [2, 0, 0], [3, 0, 2], [2, 1, 1], [0, 0, 2]], int)

Max = np.array([[7, 5, 3], [3, 2, 2], [9, 0, 2], [2, 2, 2], [4, 3, 3]], int)

Available = np.array([3, 3, 2], int)

banker = Bankers(Allocation, Max, Available)

banker.is\_safe()

if \_\_name\_\_ == "\_\_main\_\_":

main()

# Outputs

Text

Description automatically generated

# Observations

I created a class that is initialized with a current allocation matrix, a maximum matrix, and an available matrix. I then create an array representing if processes are still running, then loop while any of the processes are still running. Within that loop, I check if any processes can run with the available remaining resources if they use their max possible resources; all processes that can run are marked as done running and the resources that the process uses are added to the available resources. If no processes can run, the loop breaks and the program prints that no sequence is possible. If all processes can run, the program prints out the sequence in which they ran. I based this off the version of the banker’s algorithm found on Wikipedia: <https://en.wikipedia.org/wiki/Banker%27s_algorithm>.

It is a relatively simple algorithm, especially since it is essentially a brute force solution to find just any solution, not the best. Writing it out myself made it very clear how it worked.