LẬP TRÌNH TOÁN SONG SONG

TES1

Code: Giải cứu bệnh nhân

import multiprocessing

import random

import time

from tabulate import tabulate

# Hàm sinh số bệnh nhân với đệ quy

def generate\_floor\_patients(*M*, *total*, *max\_val*=10):

    if *M* == 1:

        return [*total*] if *total* <= *max\_val* else None

    valid\_choices = [v for v in range(0, min(*max\_val*, *total*) + 1)]

    random.shuffle(valid\_choices)

    for v in valid\_choices:

        sub = generate\_floor\_patients(*M* - 1, *total* - v, *max\_val*)

        if sub is not None:

            return [v] + sub

    return None

# Hàm kiểm tra xem còn bệnh nhân để cứu ở tầng hiện tại không

def has\_patients\_in\_current\_floor(*hotel*, *room\_flood\_status*, *floor\_id*):

    for room in range(len(*hotel*[*floor\_id*])):

        if not *room\_flood\_status*[*floor\_id*][room] and *hotel*[*floor\_id*][room] > 0:

            return True

    return False

# Hàm kiểm tra xem còn bệnh nhân để cứu ở bất kỳ tầng nào không

def has\_patients\_to\_rescue(*hotel*, *room\_flood\_status*, *total\_floors*):

    for floor in range(*total\_floors* - 1):

        for room in range(len(*hotel*[floor])):

            if not *room\_flood\_status*[floor][room] and *hotel*[floor][room] > 0:

                return True

    return False

# Hàm kiểm tra điều kiện dừng

def should\_stop(*hotel*, *room\_flood\_status*, *nurses*, *roof\_nurse\_count*, *total\_floors*):

    no\_patients = not has\_patients\_to\_rescue(*hotel*, *room\_flood\_status*, *total\_floors*)

    all\_nurses\_on\_roof = sum(*nurses*[:-1]) == 0 and *roof\_nurse\_count*.value > 0

    return no\_patients and all\_nurses\_on\_roof

# Hàm quản lý từng tầng - Di chuyển khi hết bệnh nhân ở tầng hiện tại

def floor\_process(*floor\_id*, *num\_rooms*, *x\_time*, *hotel*, *nurses*, *room\_flood\_status*,

*flood\_status*, *roof\_patient\_count*, *roof\_nurse\_count*, *total\_floors*, *lock*, *done\_event*):

    while not all(*room\_flood\_status*[*floor\_id*]):  # Chưa ngập hết tầng

        with *lock*:

            current\_nurses = *nurses*[*floor\_id*]

        if current\_nurses == 0:  # Không còn y tá thì thoát

            break

        for \_ in range(current\_nurses):

            rescued = False

            for room in range(*num\_rooms*):  # Từ trái sang phải

                with *lock*:

                    if not *room\_flood\_status*[*floor\_id*][room] and *hotel*[*floor\_id*][room] > 0:

                        patients\_to\_move = min(2, *hotel*[*floor\_id*][room])

*hotel*[*floor\_id*][room] -= patients\_to\_move

                        rescued = True

                        if *floor\_id* < *total\_floors* - 2:

*hotel*[*floor\_id* + 1][*num\_rooms* - 1] += patients\_to\_move

                        elif *floor\_id* == *total\_floors* - 2:

*roof\_patient\_count*.value += patients\_to\_move

                        break

            time.sleep(*x\_time* if rescued else *x\_time* / 10)

        # Kiểm tra nếu không còn bệnh nhân ở tầng hiện tại

        with *lock*:

            if not has\_patients\_in\_current\_floor(*hotel*, *room\_flood\_status*, *floor\_id*):

                if *floor\_id* < *total\_floors* - 2:

*nurses*[*floor\_id* + 1] += *nurses*[*floor\_id*]

                elif *floor\_id* == *total\_floors* - 2:

*roof\_nurse\_count*.value += *nurses*[*floor\_id*]

*nurses*[*floor\_id*] = 0

                break  # Thoát vòng lặp, y tá đã di chuyển

            # Kiểm tra điều kiện dừng toàn bộ

            if should\_stop(*hotel*, *room\_flood\_status*, *nurses*, *roof\_nurse\_count*, *total\_floors*):

*done\_event*.set()

                return

    # Nếu thoát do tầng ngập hoặc hết bệnh nhân

    with *lock*:

        if *nurses*[*floor\_id*] > 0:  # Nếu còn y tá chưa di chuyển trước đó

            if *floor\_id* < *total\_floors* - 2:

*nurses*[*floor\_id* + 1] += *nurses*[*floor\_id*]

            elif *floor\_id* == *total\_floors* - 2:

*roof\_nurse\_count*.value += *nurses*[*floor\_id*]

*nurses*[*floor\_id*] = 0

        if should\_stop(*hotel*, *room\_flood\_status*, *nurses*, *roof\_nurse\_count*, *total\_floors*):

*done\_event*.set()

# Hàm quản lý lũ lụt

def flood\_controller(*num\_floors*, *num\_rooms*, *x\_time*, *room\_flood\_status*, *flood\_status*, *lock*, *done\_event*):

    current\_floor = 0

    current\_room = 0

    while current\_floor < *num\_floors* and not *done\_event*.is\_set():

        time.sleep(3 \* *x\_time*)

        with *lock*:

*room\_flood\_status*[current\_floor][current\_room] = True

            if all(*room\_flood\_status*[current\_floor]):

*flood\_status*[current\_floor] = True

                current\_floor += 1

                current\_room = 0

            else:

                current\_room += 1

# Hàm hiển thị trạng thái khách sạn

def display\_hotel\_status(*hotel*, *nurses*, *room\_flood\_status*, *flood\_status*, *time\_unit*, *roof\_patient\_count*):

    headers = ["Floor/Room"] + [f"R{r+1}" for r in range(len(*hotel*[0]))] + ["Nurses"]

    table = []

    total\_floors = len(*hotel*)

    for floor in range(total\_floors):

        if floor < total\_floors - 1:

            row = [f"Floor {floor+1} ({'Flooded' if *flood\_status*[floor] else 'Safe'})"]

            for room in range(len(*hotel*[floor])):

                cell = "x" if *room\_flood\_status*[floor][room] else f"{*hotel*[floor][room]}P"

                row.append(cell)

            row.append(*nurses*[floor])

        else:

            row = ["Roof", f"{*roof\_patient\_count*.value}P (Rescued)"] + [""] \* (len(*hotel*[0]) - 1) + [0]

        table.append(row)

    print(f"\nTime: {*time\_unit*}")

    print(tabulate(table, *headers*=headers, *tablefmt*='grid'))

# Hàm vòng lặp hiển thị

def display\_loop(*done\_event*, *hotel*, *nurses*, *room\_flood\_status*, *flood\_status*, *x\_time*, *lock*, *roof\_patient\_count*):

    t = 0

    while not *done\_event*.is\_set():

        with *lock*:

            display\_hotel\_status(*hotel*, *nurses*, *room\_flood\_status*, *flood\_status*, t, *roof\_patient\_count*)

        time.sleep(3 \* *x\_time*)

        t += 3 \* *x\_time*

# Hàm chính

def main():

    # Tham số

    N = 5  # Số tầng (bao gồm mái)

    M = 8  # Số phòng mỗi tầng

    X = 1  # Đơn vị thời gian X

    manager = multiprocessing.Manager()

    lock = multiprocessing.Lock()

    # Khởi tạo khách sạn

    hotel = manager.list()

    for \_ in range(N - 1):

        floor\_patients = generate\_floor\_patients(M, M \* 3, *max\_val*=10)

        if floor\_patients is None:

            floor\_patients = [3] \* M

        hotel.append(manager.list(floor\_patients))

    hotel.append(manager.list([0]))  # Tầng mái

    # Khởi tạo nurse

    nurses = manager.list([M // 4 for \_ in range(N - 1)] + [0])

    initial\_nurses = sum(nurses)

    total\_patients = sum(sum(floor) for floor in hotel[:-1])

    # Khởi tạo trạng thái ngập

    room\_flood\_status = manager.list([manager.list([False] \* M) for \_ in range(N - 1)] +

                                     [manager.list([False])])

    flood\_status = manager.list([False] \* N)

    roof\_patient\_count = multiprocessing.Value('i', 0)

    roof\_nurse\_count = multiprocessing.Value('i', 0)

    done\_event = manager.Event()

    # Tạo tiến trình

    processes = [

        multiprocessing.Process(*target*=flood\_controller,

*args*=(N - 1, M, X, room\_flood\_status, flood\_status, lock, done\_event))

    ]

    processes.extend(multiprocessing.Process(*target*=floor\_process,

*args*=(i, M, X, hotel, nurses, room\_flood\_status,

                                                  flood\_status, roof\_patient\_count, roof\_nurse\_count, N, lock, done\_event))

                    for i in range(N - 1))

    # Tạo tiến trình hiển thị

    p\_display = multiprocessing.Process(*target*=display\_loop,

*args*=(done\_event, hotel, nurses, room\_flood\_status, flood\_status, X, lock, roof\_patient\_count))

    # Bắt đầu tiến trình

    for p in processes:

        p.start()

    p\_display.start()

    # Chờ hoàn thành

    for p in processes:

        p.join()

    done\_event.set()

    p\_display.join()

    # Tính kết quả

    dead\_patients = sum(hotel[f][r] for f in range(N - 1) for r in range(M)

                        if room\_flood\_status[f][r])

    saved\_patients = roof\_patient\_count.value

    remaining\_nurses = roof\_nurse\_count.value

    dead\_nurses = initial\_nurses - remaining\_nurses

    efficiency = (saved\_patients / total\_patients) \* 100 if total\_patients > 0 else 0

    # In kết quả

    print("\n--- Final Results ---")

    print(f"Tổng số bệnh nhân ban đầu: {total\_patients}")

    print(f"Số bệnh nhân được cứu: {saved\_patients}")

    print(f"Số bệnh nhân đã chết: {dead\_patients}")

    print(f"Tổng số y tá ban đầu: {initial\_nurses}")

    print(f"Số y tá còn lại (sống sót): {remaining\_nurses}")

    print(f"Số y tá đã chết: {dead\_nurses}")

    print(f"Hiệu suất cứu bệnh nhân: {efficiency:.2f}%")

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

    main()

Kết quả cuối cùng:

Tổng số bệnh nhân ban đầu: 96

Số bệnh nhân được cứu: 96

Số bệnh nhân đã chết: 0

Tổng số y tá ban đầu: 8

Số y tá còn lại (sống sót): 8

Số y tá đã chết: 0

Hiệu suất cứu bệnh nhân: 100.00%  
Quá trình cứu bệnh nhân:

A screenshot of a computer

AI-generated content may be incorrect.

.

.

.

A table with text and numbers

AI-generated content may be incorrect.