DAA Assignment

By Shravani Ankur Wanjari Class : CSE-B Student ID: 22WU0101093

Assignment 1.

Hogwarts has invited N teachers to educate people about the COVID-19 pandemic. The Ministry of Magic has decided to announce a total lockdown in Hogwarts, if the condition doesn't improve in next D days. So, the teachers have got only D days to teach. At most one lecture can be scheduled each day.

The ith teacher arrives on day Di and stays till lockdown. He/She also wants to teach exactly Ti lectures.

For each lecture that a teacher was not able to teach, he/she will curse Hogwarts and the curse level will increase by Si.

You have been assigned the job to schedule lectures so that Hogwarts will have to face minimum curse after D days

Solution:

```
def sort_by_curse(teacher):
  return teacher[2]

total_teachers, total_days = map(int, input().split())

all_teachers = []

for _ in range(total_teachers):
  arrival, lectures, curse = map(int, input().split())
  all_teachers.append((arrival, lectures, curse))

def minimize_total_curse(total_teachers, total_days, all_teachers):

  all_teachers.sort(key=sort_by_curse, reverse=True)

  schedule = [0] * (total_days + 1)

  total_curse = 0
```

```
  for arrival, lectures, curse in all_teachers:
    for day in range(arrival, total_days + 1):
      if schedule[day] == 0 and lectures > 0:
        schedule[day] = 1
        lectures -= 1
  total_curse += lectures * curse
  return total_curse
print(minimize_total_curse(total_teachers, total_days, all_teachers))
```

Input and Output

Input	Output
2 3 1 2 300 2 2 100	100
2 3 1 1 100 2 2 300	0
2 3 3 2 150 1 1 200	150

Result : Standard input output achieved

Note: Didn’t check for time complexity and space complexity because figuring out the code was too much work itself

Assignment - 2

Dr Strange v/s COVID-19 - (Subtask 3)
Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 256 megabytes
A COVID-19 vaccine has finally been found but it’s in a different universe. There are a total of n universes in this multiverse and Dr Strange can make portals between any of
them. They are numbered 1 to n and travelling between any two is only

Dr Strange is too important to leave Earth right now, so he sends Spiderman instead to hop from universe to universe.
Also, some of the universes are captured by Dark Forces and are patrolled by demons at specific time instances. In order to pass through them, Spiderman has to hide there until it’s safe. If Spiderman arrives
at a universe at time t and it is marked unsafe at time t, then Spiderman would

possible through these specially created portals. Earth is in the 1st universe whereas the vaccine is known to be on the nth universe.

Due to the complexity of multiverse space-time continuum, different portals can take an unequal amount of travelling time (in minutes).

have to hide until the next safe minute arrives.
Since Earth is in dire need of the vaccine, Dr Strange decides to get it at the earliest. With information about the portal travel time and unsafe minutes for each universe, help Dr. Strange make the fastest route
for Spiderman to take, in order to reach the vaccine.

Solution:

```
def find_min_time(total_nodes, total_edges, graph, traffic):
  queue = [(0, 1)] # [(time, node)]
  visited = set()
  while queue:
    queue.sort()
    time, node = queue.pop(0)
    if (time, node) in visited:
      continue
    visited.add((time, node))
    if node == total_nodes:
      return time
    for start, end, weight in graph:
      if start == node:
        new_time = time + weight
        while new_time in traffic[end]:
          new_time += 1
        queue.append((new_time, end))
total_nodes, total_edges = map(int, input().split())
```

```
graph = []

for _ in range(total_edges):

  edge = tuple(map(int, input().split()))

  graph.append(edge)

traffic = {}

for node in range(1, total_nodes+1):

  jam_times = list(map(int, input().split()[1:]))

  traffic[node] = jam_times

min_time = find_min_time(total_nodes, total_edges, graph, traffic)

print(min_time)
```

Input and Output

Input	Output
4 4 1 2 3 1 3 2 2 4 2 3 4 3 0 1 4 2 2 3 0	5

Result: Output Achieved with the given input