

ALGORITHM DESIGN PROJECT

1193. Queue at the Exam

Difficulty: 178

by

Thanapol Sopanhari ID: 6113632

Problem Definition

A group of students are having an oral examination. At the beginning of the exam, all students simultaneously receive their questions and start preparing for the answer. Each student needs T_1 minutes for the preparation and T_2 minutes for the answer itself (these parameters can be different for different students). For each student, the time T_3 (in minutes from the planned beginning of the exam) is given when this student must be free because she has other things to do (for example, other exams).

During the exam, a queue of students is formed as they are getting ready to speak. If a student is ready to answer and at that time moment the professor is free, then this student starts answering immediately. If the professor is busy with another person, then the student joins the queue and starts answering when the student before her in the queue finishes her examination.

It is possible that some students won't be free when they planned to be (i.e., at the time T_3). The professor is ready to cooperate and can shift the beginning of the exam to an earlier time. However, he doesn't want to come too early! Your task is to write a program that will calculate the minimal period in minutes by which the exam should be shifted so that all the students will manage to complete the exam before their T_3 time.

Problem Definition

Input

The first line contains the number of students N ($1 \leq N \leq 40$). Each of the next N lines contains the corresponding numbers T_1 , T_2 , and T_3 . The numbers are separated with spaces and satisfy the constraints $0 \leq T_1 \leq T_3 \leq 600$, $1 \leq T_2 \leq 240$. All the numbers T_1 are distinct.

Output

Output the nonnegative integer that is the answer to the problem. If there is no need to shift the beginning of the exam, output 0.

Problem Definition

Sample

| Input | Output |
|---|--------|
| 3 100 10 120 70 40 150 99 15 400 | 15 |
| 2 100 10 110 80 15 100 | 0 |

Problem Analysis

- ◇ T_1 minutes for the preparation
- ◇ T_2 minutes for the answer itself
- ◇ T_3 minutes for a student must be free for doing other things.
- ◇ A student should start answering as fast as when a student and the teacher are ready by ascending order of T .
- ◇ The next student must starts answering immediately when the previous student finished the examination(t_2).
- ◇ If T_1 plus T_2 is more than T_3 and the different between T_3 and $T_1 + T_2$ is more than the previous shift time, the new shift time will be the time difference.
- ◇ If the next student is ready, the sum of previous students will be plus with the minutes for answering of the next student.
- ◇ If the previous student finished answering, but the next student isn't ready because of the preparation, the professor must wait for him/her.
- ◇ If there is no need to shift the beginning of the exam, the shift time is 0.

Problem Analysis

| Input |
|------------|
| 3 |
| 100 10 120 |
| 70 40 150 |
| 99 15 400 |



| Solution |
|--|
| $70 + 40 = 110$ (70 40 150) |
| $110 + 15 = 125$ (99 15 400) |
| $125 + 10 = 135$ (100 10 120) |
| The required time is 135 which is more than 120, the shift time is 15 (135 - 120). |

Problem Solution

- ◇ Programming language : Python 3.5
- ◇ Firstly, I created the function called getKey for getting the position.
 - ◇ time[0] means the function will return first item in the list which is index t1.

```
def getKey(time):  
    return time[0] # t1
```

- ◇ Secondly, I created a variable N to get the number of students and created a list of times which will be used to store inputs. After that, I looped with N times to add line i containing 3 variables which are t1, t2, t3. A list of times will be 2-dimension array.

```
N = int(input())  
times = []  
  
for i in range(N):  
    time = list(map(int, input().split()))  
    times.append(time)
```

Problem Solution

- ◇ Next, I declared 2 variables:
 - ◇ 1. Storing times used for each student.
 - ◇ 2. Storing times that a teacher need to change an exam time for all students can be free on time(t3).

```
useT = 0  
shiftT = 0  
  
times.sort(key=getKey)
```

- ◇ I sorted a list of times by setting the key with getKey function. The sort will be based on the first index which is t1. So that, professor can know which student will be the next one, as a queue.

Problem Solution

- ◇ Then, I looped with N times.
- ◇ Inside the loop, I checked if useT is lower than the first index of times i. If it's true, useT will be equal to the first index of times i (t1).
- ◇ After the first condition finished, useT plus with the second index of times i which is t2. needT will be equal to subtraction between useT and the third index of times i which is t3.
- ◇ Lastly, I checked whether needT is more than shift. If it's true, shiftT will be equal to needT.
- ◇ When the loop ended, I printed out the result that professor needs to shift the beginning of the exam.

```
for i in range(N):  
    if useT < times[i][0]:  
        useT = times[i][0]  
    useT += times[i][1]  
  
    needT = useT - times[i][2]  
    if needT > shiftT:  
        shiftT = needT  
  
print(shiftT)
```

Source Code

```
def getKey(time):  
    return time[0] # t1  
  
N = int(input())  
times = []  
  
for i in range(N):  
    time = list(map(int, input().split()))  
    times.append(time)  
  
useT = 0  
shiftT = 0  
  
times.sort(key=getKey)  
for i in range(N):  
    if useT < times[i][0]:  
        useT = times[i][0] + times[i][1]  
    else:  
        useT += times[i][1]  
  
    needT = useT - times[i][2]  
    if needT > shiftT:  
        shiftT = needT  
  
print(shiftT)
```

Submission Result

Solutions judgement results

| ID | Date | Author | Problem | Language | Judgement result | Test # | Execution time | Memory used |
|---------|-------------------------|-------------------------|---|----------------|------------------|--------|----------------|-------------|
| 9035237 | 09:32:54 29 Sep 2020 | balloon | 1193. Queue at the Exam | Python 3.8 x64 | Accepted | | 0.093 | 320 KB |

Thank you