HOMEWORK 3 – Q3

MINGLANG XIE

z5228006

3. You are given a time schedule of arrivals $\mathrm{a_i}$ and departures $\mathrm{d_i}$ of n trains, so $1 \leq i \leq n$, during each 24 hour period (note: a train can arrive before the midnight and leave after midnight; each train arrives and departs at the same time every day). You need to find the minimum number of platforms so that each train can stay at a platform without interfering with other arrivals and departures.

Solution:

Assume there is no train can be stay at a platform for more than 24 hours, such as 24:00-24:01 means a train stay in a platform for 1min, and 24:00-24:00 means a train stay in a platform for 24 hours, then departs at the same time a train arrive, this does not make a conflict.

First, go through the timetable schedule of arrivals and departures and count how many trains arrive before midnight and depart after midnight. This number is the initial value of your counter.

Second, make a single list of the departures and arrivals times (labeled as S) sort such list in an increasing sequence of times. For example, we have two trains, a train's arrivals time is 22:00, and departures time is 1:00; the other train's arrivals time is 2:00, and departures time is 3:00. We make a sequence $\langle(1{:}00, depart), (2{:}00, arrive), (3{:}00, depart), (22{:}00, arrive)\rangle$, we then go through this sequence, and record the maximum counter during the iteration. When we have a 'depart', we decrease the counter, and increase the counter when we have a 'arrive'. The maximum counter during the iteration is the minimum number of platform so that each train can stay at a platform without interfering with other arrivals and departures.

Time complexity: Sorting all trains takes at most $O(n\log(n))$ time. S have $2n$ elements, so go through the sequence S will be at $O(n)$ time. Therefore, the time complexity is $O(n\log(n) + n) = O(n\log(n))$

```
counter = 0
for train in trains:
    if train arrive before midnight and depart after midnight:
        counter += 1

S = []
for train in trains:
    S += (train.arrivalTime, ‘arrival’)
    S += (train.departureTime, ‘depart’)
```

```
Merge-Sort(S) by time

Answer = 0
for time, state in S:
    if state is depart:
        counter -= 1
    else if state is arrival:
        counter += 1
    if counter > answer:
        answer = counter
```