

Question 3

This is basically a question of activities selection with multiple resources. Here, we trend train station as a resource and train as a request with certain departure time and arrival time.

But one thing is different which is the train can arrive and departure across midnight. We need to break the time interval and trend it as two different requests. That is, if a train i departure at time s and arrive at time f and time interval $[s, f]$ across midnight, we split request i to be i_1 and i_2 which have time intervals $[i_1, \text{midnight}]$, $[i_2, \text{midnight}]$.

Then, we have $n + m$ resources where m is the number of trains which across midnight. Now we need to find the maximum number of overlapping interval because this is the minimum possible number of train stations.

Overall, the algorithm could be described as follow.

1. Get the updated request number.
2. Find the maximum number of overlapping

Pseudo code:

```
/* get the correct train number */
for t in train:
    if t.endtime > midnight:
        n ++;          /* train number */

/* find the maximum overlapping */

/* array contain the start time and end time in increasing order */
start_time[] = sort_time(train.starttime)
end_time[] = sort_time(train.endtime)

/* init a counter. when we find a interval, inc the counter. When a
interval comes to end, dec the counter.
* if there is not overlapping, the value of counter should be 0
* the counter would contain the maximum number of overlapping
*/
i, j = 0
while (i < n && j < n) {
    if (start_time[i] < end_time[j]) {      // find interval
        counter ++;
        i ++;                             // next start time
    }
    else {
        counter --;                        // end of interval
        j ++;                             // next end time
    }
}
```

}
}

The maximum overlapping number is the minimum possible number of train stations.