**Yuan Gao z5239220 Q3**

You are given a time schedule of arrivals ai and departures di of n trains, so , 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.

Assume time schedule accurate to minutes

Therefore, the whole day can be seen as discrete points ( = 1440min)

Create a Two-dimensional array , every element contain two numbers which are departures and arrivals.()

Step 1:

Separate the schedule which across midnight to the two schedules.

and

Convert a cycle to a line

(if means across midnight)

Step2:

Sort array based on time

Step3:



We need to calculate the maximum number of trains stay in platform at same time. Only in this way can there be no conflicts.

It makes sure every train has platform to stop

Code:

1. **int** findMinplatform(vector<vector<**int**>>& timeTable){
2. **if**(timeTable.size() < 1) **return** timeTable.size();
4. //Separate the schedule which across midnight to the two schedules.
5. auto i = timeTable.begin();
6. **while**(i != timeTable.end()){
7. **if**((\*i)[0] > (\*i)[1]){
8. vector<**int**> b = {0, (\*i)[1]};
9. (\*i)[1] = 1044;
10. i = timeTable.insert(i + 1, b);
11. }**else**{
12. ++i;
13. }
14. }
16. //Sort timeTable array based on departures time
17. sort(timeTable.begin(), timeTable.end(), [](vector<**int**> &a, vector<**int**> &b){
18. **return** a[0] < b[0];
19. });
21. //calculate the maximum number of trains stay in platform at same time
22. //initialize the first interval
23. **int** left = timeTable[0][0];
24. **int** right = timeTable[0][1];
25. //initialize the result and tmp
26. //res is the minimum number of platforms
27. **int** res = 1;
28. //tmp is count the maximum number of trains stay in platform at same time
29. **int** tmp = 1;
31. **for**(**int** i = 1; i < timeTable.size(); ++i){
32. //if current departures time not in the interval
33. //picked max(res, tmp)
34. **if**(right < timeTable[i][0]){
35. left = timeTable[i][0];
36. right = timeTable[i][1];
37. res = max(res, tmp);
38. tmp = 1;
39. }**else**{
40. //if current departures time in the interval
41. //shrink the interval
42. **if**(left < timeTable[i][0]){
43. left = timeTable[i][0];
44. }
45. **if**(right > timeTable[i][1]){
46. right = timeTable[i][1];
47. }
48. //the number of trains stay in platform at same time +1
49. ++tmp;
50. }
51. }
52. **return** res;
53. }