**Maximum trains for which stoppage can be provided**

**Difficulty: MEDIUM**

**Problem Statement**

**You are given the ‘n’ number of trains and ‘m’ number of a platform for a station. Every train has an associate with ‘arrival time’, ‘departure time’, and ‘platform’ number.**

**Your task is to determine the maximum number of trains for which you can provide a stoppage at the station.**

**You can provide stoppage to only one train at platform ‘x’ between ‘arrival time’ to ‘departure time’ of the current train.**

**If ‘arrival time’ and ‘departure time’ is 1015 then consider as 10:15.**

**Input Format:**

The first line of input contains an integer ‘T’ denoting the number of test cases.

The first line of each test case contains two space-separated integers ‘n’ and ‘m’, where ‘n’ denotes the number of trains and ‘m’ denotes the number of platforms.

Next ‘n’ lines contain three space-separated integers ‘arrival time’, ‘departure time’, and ‘platform’ number.

**Output Format:**

For every test case, return the maximum number of trains for which stoppage can provide.

Output for each test case will be printed in a separate line.

**Note:**

You do not need to print anything; it has already been taken care of. Just implement the given function.

**Constraint :**

1 <= T <= 100

1 <= N <= 3000

1 <= M <= 3000

0000 <= Arrival Time <= 2359

0000 <= Departure Time <= 2359

1 <= Plateform Number <= M

Where ‘T’ represents the number of test cases, ‘N’ is the number of trains, ‘M’ is the number of platforms, and ‘Arrival Time’ and ‘Departure Time’ is ‘HH:MM’ time of arrival and departure time of trains.

**Time Limit: 1 sec**

**Sample Input 1:**

2

5 2

0950 1005 1

1000 1030 1

1015 1030 1

1200 1205 2

1215 1230 2

4 1

1200 1210 1

1205 1220 1

1215 1230 1

1215 1240 1

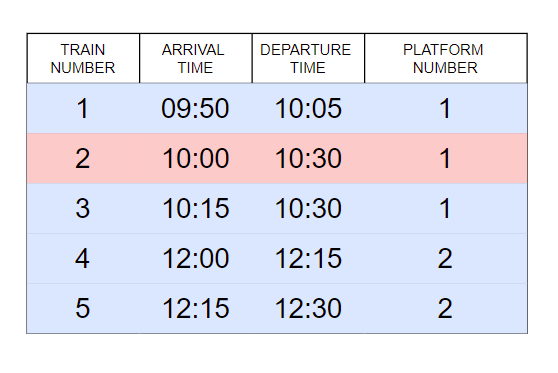
**Sample Output 1:**

4

2

**Explanation Of Sample Input 1:**

Test Case 1:



‘Platform number = 1’,

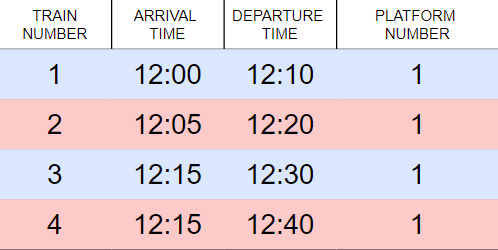
You can provide stoppage to at max ‘2’ train in-between ‘09:50’ to ‘10:30’. If you can give stoppage to train number ‘2’, only one train can be provided in-between ‘10:00’ to ‘10:30’ so provide stoppage to train number ‘1’ and ‘3’.

‘Platform number = 2’,

You can provide stoppage to both trains ‘4’ and ‘5’.

Return ‘ 2 (platform-1)+ 2 (platform-2) =4’.

Test Case 2:



‘Platform number = 1’,

You can provide stoppage to only two trains either ( ‘1’, ‘3’ ) or ( ‘1’, ‘4’ ).

Return ‘2’.

**Sample Input 2:**

2

3 1

1000 1030 1

1050 1100 1

1100 1130 1

3 1

1000 1010 1

1000 1020 1

1000 1030 1

**Sample Output 2:**

3

1

#include <bits/stdc++.h>

bool cmp(pair<int, int> p1, pair<int, int> p2) {

    if (p1.second != p2.second) return p1.second < p2.second;

    else return p1.first < p2.first;

}

int maxStop(vector<vector<int>> &trains, int n, int m) {

    // Write your code here

    vector<vector<pair<int, int>>> a(m+1);

    for (int i=0; i<n; i++) {

        a[trains[i][2]].push\_back({trains[i][0], trains[i][1]});

    }

    for (int i=1; i<=m; i++) {

        if (!a[i].empty()) sort(a[i].begin(), a[i].end(), cmp);

    }

    int ans = 0;

    for (int i=1; i<=m; i++) {

        if (!a[i].empty()) {

            vector<int> dp(a[i].size(), 0);

            dp[0] = 1;

            for (int j=1; j<a[i].size(); j++) {

                int l = 0, r = j-1, last = -1;

                while (l <= r) {

                    int mid = (l+r)/2;

                    if (a[i][mid].second <= a[i][j].first) {

                        last = mid;

                        l = mid+1;

                    }

                    else r = mid-1;

                }

                if (last != -1) dp[j] = max(dp[j-1], dp[last]+1);

                else dp[j] = dp[j-1];

            }

            ans += dp[a[i].size()-1];

        }

    }

    return ans;

}