COP 3503 Recitation #6 Problem: Greedy Algorithms (Week of 6/24)
Due: 7/4/13 (Thursday) at 11:55 PM Webcourses2 time

Problem: Catching Trains

You are at train station #0 and must get to train station #10. Trains always run from lower station numbers to the next higher station number. Thus, all trains always stop at each station. Assume that you can instantaneously get off one train at a station and board another. You are in a race with someone else to get to station #10. If you happen to tie, then the tie breaker is broken as follows: your penalty time for each train station is the number of minutes after you started at station #0 that you arrived at that station. The sum of the penalty times for stations #1 through #9 is the number of penalty points you accrue. Whoever accrues the least number of penalty points wins. For example, if you arrived at station #1 10 minutes after you started, station #2 20 minutes after you started, station #3 30 minutes after you started, etc. you would get to station #10 in 100 minutes and accrue 450 penalty points in the process. Given an input file with a complete train schedule, determine the amount of time (in minutes) necessary to get to station #10 as well as the least number of penalty points you can accrue in the process. If it is impossible to reach station #10 with the given train schedule state this. Assume that you start at station #0 at time=0 minutes.

Input File Format (trains.in)

The first line of the input file will contain a single positive integer, *n*, representing the number of train schedules in the file. Each of these schedules follows. Each train schedule will consist of 10 sets of data. The kth set of data represents information for trains from station #(k-1) to station #k. Each set of data starts with a line with a single positive integer *t*, representing the number of trains that run from station #(k-1) to station #k. The next *t* lines will contain information about the *t* trains, with information about one train on one line. The information about a single train is two integers, *s* and *f*, the start and finish times of that particular train. These two values will be separated by an integer on a line. Here is a sample file:

```
1
1
10 20
2
10 30
20 40
1
45 50
1
55 60
1
60 65
1
65 70
2
```

```
70 75
72 74
1
74 80
1
81 90
1
92 100
```

Output Format

Precede the output for each case with the following header:

```
Train A:
```

where A ($1 \leq A \leq n$) represents the one-based input case number.

If station #10 is reachable for the given schedule, output a statement with the following format, on the same line:

```
Arrive at station #10 in X minutes with Y penalty points.
```

where X is the minimum number of minutes it takes to arrive at station #10, and Y is the minimum number of penalty points to do so.

If station #10 is NOT reachable, output the following:

```
Station #10 is NOT reachable.
```

For our example above, the correct output is:

```
Train 1: Arrive at station #10 in 100 minutes with 549
penalty points.
```

(Note: This output should be on a single line. Separate output for separate cases with a blank line.)