CS 602

Algorithm Design and Implementation

Assignment 4

[Question 1] No-less-than-10 Queens Puzzle

The eight queens puzzle is the problem of placing eight chess queens on an 8×8 chessboard so that no two queens threaten each other; thus, a solution requires that no two queens share the same row, column, or diagonal (including all diagonals running from upper left corner to lower right corner and running from upper right corner to lower left corner). In this question, you are to count the number of ways to place $(14 - x)$ queens on a 14×14 chessboard where the positions of $x \in \{2,3,4\}$ queens are already known.

Test inputs begin with the number of test cases. Each test case contains pairs of positions of the x known queens. For example, “1 1 2 9 3 6 4 10” contains the positions of 4 queens, where the first queen is at first row first column, the second queen is at second row ninth column, and so on.

For each test case, output one integer representing the number of ways to replace the remaining $(14 - x)$ queens.

<u>Sample Input</u>

```
3
1 1 2 9 3 6 4 10
1 9 12 8 8 5 2 7
2 10 10 3
```

<u>Sample Output</u>

```
39
32
2414
```

[Question 2] Escape from Temapura

The new coronavirus is now spreading rapidly on the planet Temapura, many citizens are considering escaping from the planet. However, it is challenging to escape from Temapura besides it is hard and expensive to get a ticket. There are two main reasons: (1) not every city on the planet is equipped with a spaceship station, and people must drive to a city with a spaceship station to escape; (2) everyone is issued with a personal health code, a green code, an orange code, or a red code, and only those who are holding a green code issued by the local government of the city where the spaceship station is located, can board spaceships. The rules of issuing personal health codes are as follow:

- Everyone who wishes to travel is issued with a green personal health code by his or her own city government, and this code does not change till the arrival at the next city.
- Upon arrival of the next city, depending on the relationship between the two local governments, the personal health code could be downgraded if the two governments do not trust each other: a green code will be downgraded to an orange code, or an orange code will be downgraded to a red code. You may understand it as a reissuance of the health code by the government of the arrival city. We call the health code reissued at the arrival city as “arrival code”. Of course, if they two governments trust each other, the health code is not downgraded, i.e., the arrival code is the same as the original health code.
- It is not allowed to travel with a red code, however, travelling with a green code or an orange code is allowed. It is thus possible to have red arrival code when traveling from a distrusted city with an orange code.
- The health code system is certainly not implementable if there are only rules for downgrade. If the arrival code is red, the person must be quarantined to get an orange code or a green code; elif the arrival code is orange, he has the choice to be quarantined to get a green code, or proceed to another city with an orange code issued by the arrival city. At the final city with spaceship station, people must get quarantined for a green code from that local government if they arrived with an orange code or a red code.
- Note that everyone has two choices upon arrival, either choose different durations to get an orange code or a green code (when arrival code is red) or choose to get quarantined (when arrival code is orange). We do not consider the case that people choose to get quarantined with a green arrival code as crazy people are not allowed to board spaceships.
- The duration to quarantine for a green code is longer than the duration for an orange code starting from a red arrival code. For example, to quarantine for a green code takes 14 blinks while to quarantine for an orange code takes 7 blinks, where “blink” is the unit of time on planet Temapura. However, the government of each city may define such durations in the way they like, there is no standard definition. But fortunately, their maths are good enough to derive the sum of the durations from red arrival code to orange code and from orange arrival code to green code equals to the duration from red arrival code to green code.

In this question, you are given (1) the traveling time and trust relationships between cities with direct road connections in Temapura, (2) the quarantine durations in blinks for every city (two durations for each, one from red to orange, the other from orange to green), and (3) the list of cities with spaceship stations, compute the minimum blinks required to escape from Temapura (yes, you can escape from any spaceship stations, no need to consider ticket prices!) for all cities.

Your task is to implement the algorithm `escape` which computes the minimum blinks for all cities. Inputs begin with the number of cities n ($1 \leq n \leq 15{,}000$) in Temapura, this is then followed by a line describing triples of city IDs (the first two elements, city ID starts from 1 and ends at n) connected by a direct road and the number of blinks (the last element) as the traveling duration on this road. The signs of the number of blinks represent the trust relationship between city governments. If the number of blinks is positive, the two city governments trust each other, otherwise they do not trust each other. The absolute value of the number of the blinks is then the actual traveling duration. The next line describes the quarantine period in number of blinks for each city in the ascending order of city IDs. Each quarantine period is a pair of two values, where the first value is the number of blinks required to quarantine for an orange code from a red code, and the second value is the number of blinks required to quarantine for a green code from an orange code. This implies the number of blinks required to quarantine for a green code from a red code is the sum of these two values. The last line contains the list of m ($m \leq 80$) cities with spaceship stations, the traveling time from a city to its spaceship station and within any city can be ignored.

Sample Input

```
12
1:3:47 1:5:19 1:9:-64 2:4:16 2:10:25 2:11:41 3:5:11 3:10:-32 4:5:-20 4:10:29 \
4:11:11 5:7:-45 6:8:39 6:9:36 6:12:36 7:9:28 7:12:25 8:9:32 8:12:33 9:11:-40 \
9:12:15 10:11:37 11:12:-50
13:11 36:17 12:15 23:14 19:18 28:13 15:10 10:21 27:12 16:10 15:23 21:9
10
```

Note: line 2 is too long that has been broken into 3 lines.

Sample Output

```
72 25 42 29 53 123 113 119 87 0 37 97
```

Grading Rubrics:

1. There are 10 marks for each question.
2. There are 10 test cases for question 1 and 2, where the first test case is the sample input A4Q1.in and A4Q2.in respectively. You get 1 mark for each test case if your code does not produce errors including wrong answer or time limit exceeded.

Running python skeleton with sample input:

1. Open “Anaconda Prompt”
2. Go to the directory where you put the file **A4Q1.py** and **A4Q1.in**, using command **`cd`**
3. Run command **python A4Q1.py < A4Q1.in**
4. You may want to create a test input called **my_own_test.in** to design a test case for your own program, the command would then be **python A4Q1.py < my_own_test.in**
5. Same applies to Question 2, so you may run **python A4Q2.py < A4Q2.in**
6. Make sure your program read input from **`sys.stdin`**. If you find it difficult to work with **`sys.stdin`**, you might write the function that your need to modify with your own input and output, and then copy and paste to the function itself to the code skeleton provided. Please test with the command in item 3 to ensure it produces the right output.