The Great Police-and-Scooter Problem

Description

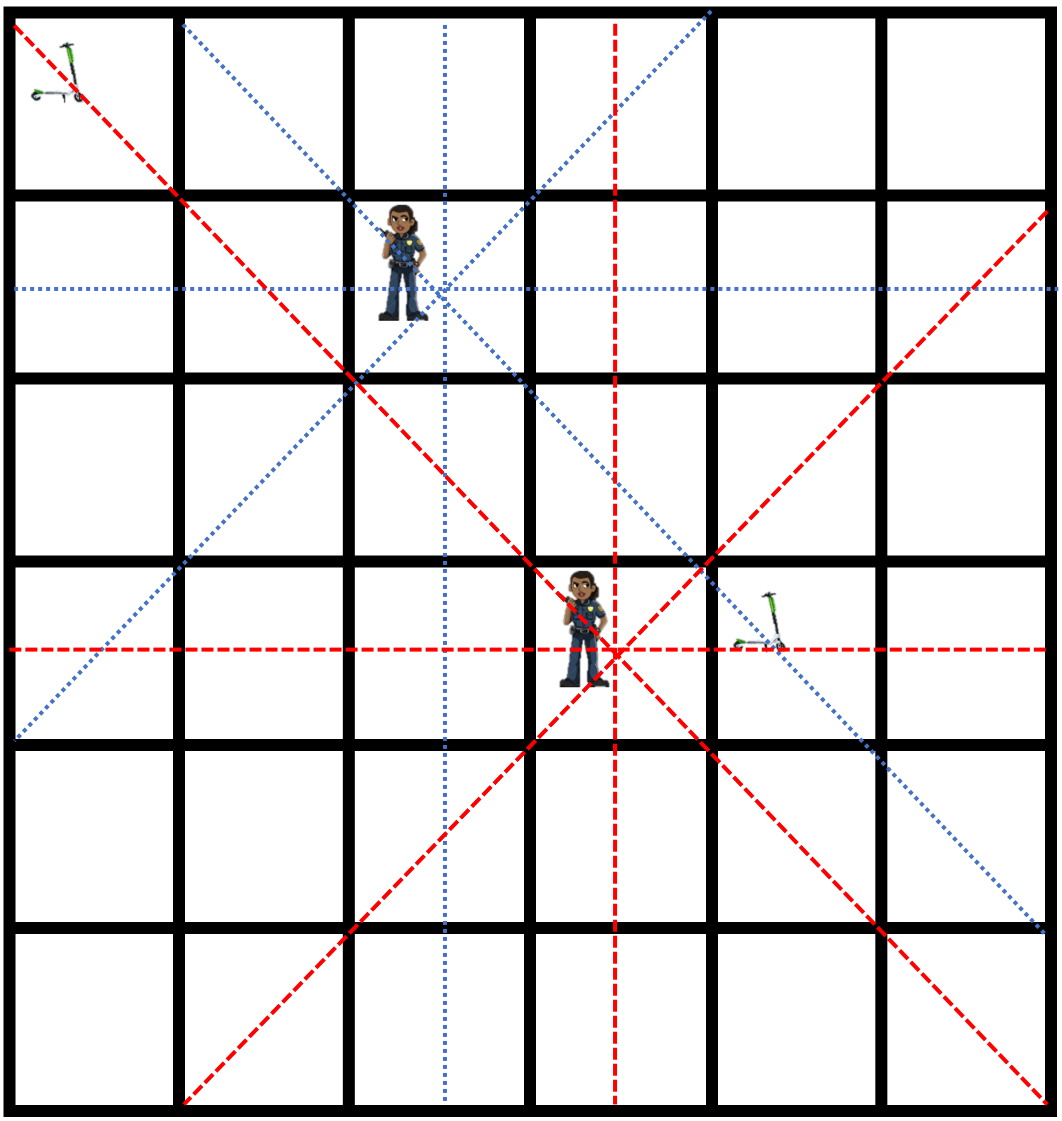
You are helping the LA Dept of Transportation (LADOT) to develop a pilot scooter program for LA. There are a limited number of police officers available to monitor and address issues that may arise from scooters, ranging from traffic and safety violations to accidents with cars, bikes, other scooters and pedestrians. The scooter companies have given LADOT access to scooter routes over the course of one day. In order to maximize the scooter activity monitored by the officers, you will take as input the route information, the monitored city area dimensions, and the number of officers available to then generate the best placement of the officers. The officers can only be in one place for one day, and there can only be one officer on each street. When an officer and scooter are at the same location at the same time, the officer is able to address a safety issue, and one "Activity point" is gained. **The goal is to place the officers in locations that do not conflict with each other, while maximizing the total "Activity points" for the day (12 time steps in a day).** The problem follows these rules:

- Officers cannot be in same square, same row, same column, or along the same diagonal. (Think of queens on a chess board)
- Officers cannot move.

- Activity points are collected at each time step t when officers are in same square as scooters. One point per each scooter.
- The grid coordinate system will be indexed starting from the top-left corner. An example of a 5 by 5 grid is given below with each cell's coordinates:

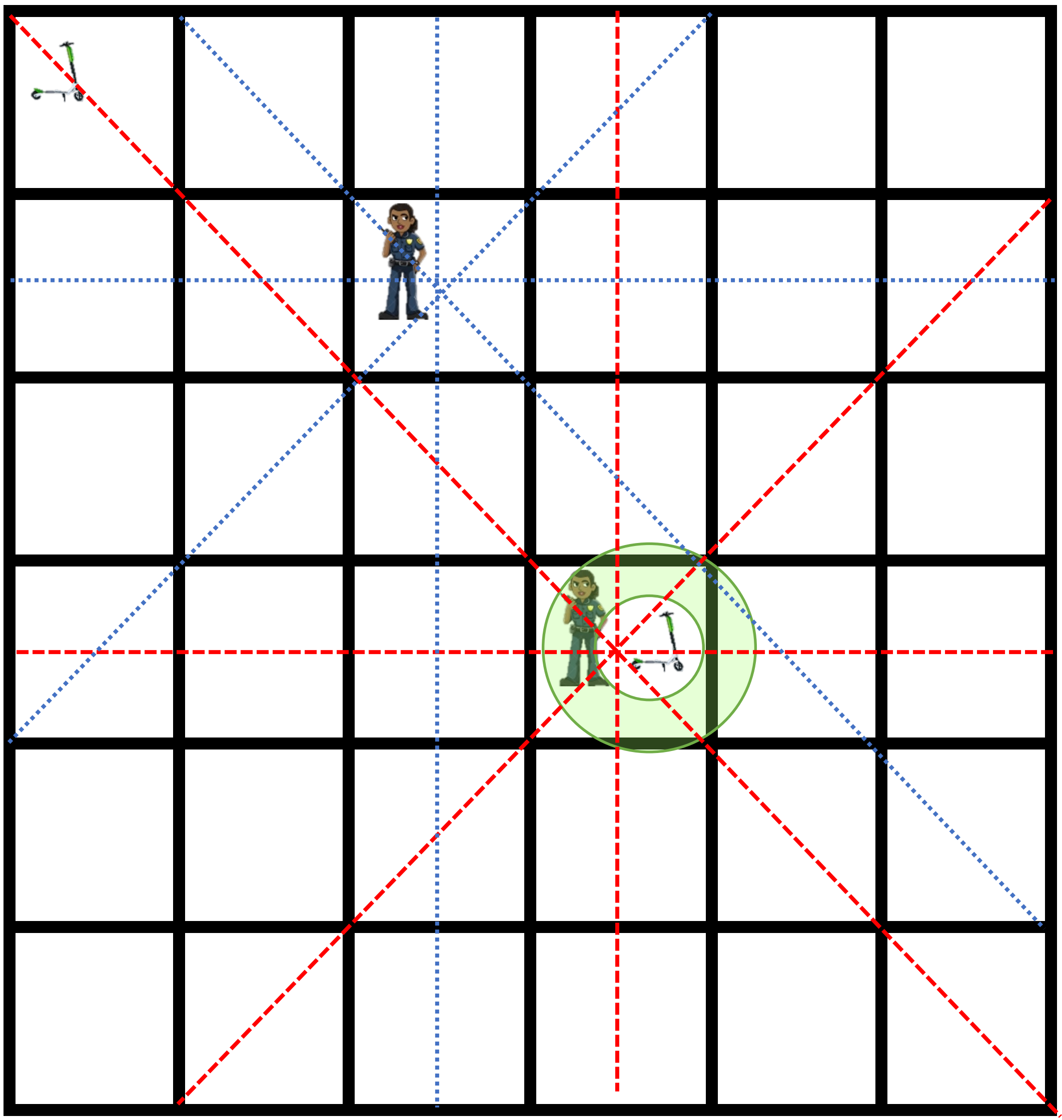
0,0	1,0	2,0	3,0	4,0
0,1	1,1	2,1	3,1	4,1
0,2	1,2	2,2	3,2	4,2
0,3	1,3	2,3	3,3	4,3
0,4	1,4	2,4	3,4	4,4

T = 1



The two police officers have been placed and are unable to move. The red and blue lines show the limitations on placing officers; no officer may be placed on the same row, column, or diagonal as another officer. Since no scooters are in the same square as either of the officers, no value is gained at T=1.

T = 2



At T=2, the scooters have both moved one square. Since a scooter has moved onto a square with an officer, 1 Activity point is gained at T=2. This process continues at T steps forward, with the officers keeping their positions.

Input: The file `input.txt` in the current directory of your program will be formatted as follows:

First line: strictly positive 32-bit integer ***n***, the width and height of the ***n*** x ***n*** city area, ***n*** <= 15.

Second line: strictly positive 32-bit integer ***p***, the number of police officers

Third line: strictly positive 32-bit integer ***s***, the number of scooters

Next ***s****12 lines: the list of scooter x,y coordinates over time, separated with the End-of-line character LF. With ***s*** scooters and 12 timesteps in a day, this results in 12 coordinates per scooter.

Output:

Max activity points: strictly positive 32-bit integer ***m***

Helpful Hints:

1. **All officers must be allocated.**
2. **Some inputs may have more than one valid solution.** We will accept all solutions that achieve the maximum possible Activity points for the given input.
3. **Think about representing the problem.**
    a. What is a good representation of states and operators?
    b. While the scooters may be moving at each timestep, the officers cannot move. How can you use this to simplify the problem representation?
    c. How will you evaluate the "score" of a state?
4. **Think about complexity.**
    a. How does grid size affect branching factor?
    b. How does the number of officers affect search tree depth?
    c. How can you use the input parameters to determine which algorithm will be able to generate a solution within 3 minutes?