LV-INTERN PROGRAMMING EVALUATION

Instructions

You have 1 day to finish this evaluation. Please attempt all the problems and try not to skip any problems completely. You can use any online resource, but cannot use AI tools like GPT or others, nor asking for help from any person.

After you have submitted your solution, **please spend 30 minutes writing a self-evaluation (in English). What did you think of the problems and how do you think you did? Send the self-evaluation via email as well.**

***Do not share this evaluation or your solutions with anyone***

Problem 1

On a square integer lattice of size N by N, we can move according to the following rules

* We always begin at (0, 0).
* We must always move first to (1, 0).
* We will move N − 1 additional steps. Each time, we can go from (a, b) to any of these four points: {(a+1,b), (a−1,b), (a,b+1), (a,b−1)}.
* We may not move to the same point twice.
* We may not move outside the lattice.

Your goal is to implement the solveProblem1() function that takes N as input, and outputs the number of unique routes that we can take through a lattice of size N.

As an example, the following is all unique routes for N = 2,3,4. The answers are f(2) = 2, f(3) = 5, f(4) = 12. Your solution does not need to generate images.

图片包含 游戏, 文字, 游戏机, 水

描述已自动生成

We will test your program with N = {5, 6, 7, ..., 24, 25}. The evaluation of each value of N should take less than 1 second.

Problem 2

Run the following Python code:

|  |
| --- |
| from functools import reduce; f = lambda m,n: sorted(list(filter(lambda a: a in range(m,n), list(reduce((lambda x,y: x - set(range(y\*\*2,n,y)) if (y in x) else x), range(2,int(n\*\*0.5) + 1), set(range(2,n))))))) |

Part 1: What does this code output? Please explain this algorithm.  
Part 2: Can you explain the filter and reduce function in python?  
Part 3: Rewrite the code to easy to understand. Add at least 5 comments.

Problem 3

”Circular Queue” is a linear data structure in which the operations are performed based on FIFO (First In First Out) principle and the last position is connected back to the first position to make a circle. (see figure below)

图表

描述已自动生成

The attached problem3.py is trying to implement the ”Circular Queue” data structure with Python. But it has four bugs and can not run correctly, please fix them and add a comment to each change you make.