Grid Path Searching

A two dimensional grid has M horizontal rows and N vertical columns. Let (i, j) denote the square at the $(i+1)^{th}$ row from the top and the $(j+1)^{th}$ column from the left i.e., they are 0-indexed.

Within the grid, for each i and j ($0 \leq i \leq M-1$, $0 \leq j \leq N-1$), there can only be two symbols `#` and `.` where `#` denotes a blockage and `.` denotes a passable opening. Starting at the upper left and only moving downwards and rightwards, find the number of connected paths between the top-left square and the bottom right square by traversing only the intermediate squares with the `.` symbol. The start and end positions are never be marked with `#`.

Example:

$M = 3$ and $N = 4$.

```
...#
.#..
....
```

calling `count_paths(3,4,[(0,3),(1,1)])` returns the answer is `3`. Here is the function signature: `count_paths(m,n,blocks)` where `m` is the number of rows and `n` is the number of columns. The `blocks` variable is a list of tuples indicating the blocked `#` entries in the grid.

Please put your Python code in a Python script file and upload it. Please retain your submitted source files! Remember to use all the best practices we discussed in class. You can use any module in the Python standard library, but third-party modules (e.g., Numpy, Pandas) are restricted to those **explicitly** mentioned in the problem description.

Tips:

- After you have submitted your file, do **not** use the browser back or reload buttons to navigate or open the page in multiple browser tabs, as this may cause your `attempts` to decrease unexpectedly. It may take up to thirty seconds for your code to be processed, so please be **patient**.
- If you find yourself back at the main page without any feedback or change in your `attempts` then it means that your code timed out or crashed in some unexpected way.
- Ensure that your development environment does not presume the existence of certain packages for the autograder. The autograder does not have anything other than the standard library and those third-party libraries **explicitly** named in the problem description.
- Do not leave extraneous statements in your code like test cases, print statements, or anything else besides what is needed to evaluate your submission because the the autograder will spend its limited time executing those lines, which may result in unexpected crashes or timeouts.

浏览... 0801.py Upload Python source code file

Correct! Back to assignments. functional points = 20 /20 and validation points = 10/10