Daily Coding Problem #166

Problem

This problem was asked by Uber.

Implement a 2D iterator class. It will be initialized with an array of arrays, and should implement the following methods:

- `next()`: returns the next element in the array of arrays. If there are no more elements, raise an exception.
- `has_next()`: returns whether or not the iterator still has elements left.

For example, given the input [[1, 2], [3], [], [4, 5, 6]], calling `next()` repeatedly should output `1, 2, 3, 4, 5, 6`.

Do not use `flatten` or otherwise clone the arrays. Some of the arrays can be empty.

Solution

To minimize the amount of space we use for the 2D iterator, we just need to store two pointers, an outer pointer `i` and an inner pointer `j`. To make the code reusable, we'll implement a private function `_next_coords(i, j)` that gets the coordinates or the next valid element.

Remember that we need to check for the following cases:

- List of lists is empty
- First element hasn't been retrieved yet
- Empty list in beginning, middle, or end
- End of inner list

```
class Iterator:
    def __init__(self, lsts):
        self._lsts = lsts
        self.i = None
        self.j = None

    def _next_coords(self, i, j):
        '''Gets the coordinates of the next valid element.'''
        if not self._lsts or len(self._lsts) == 0:
            return None

        if i is None and j is None:
            i = 0
            while i < len(self._lsts):
                if len(self._lsts[i]) > 0:
                    return (i, 0)
                i += 1

        if j + 1 < len(self._lsts[i]):
            return (i, j + 1)

        i += 1
        while i < len(self._lsts):
            if len(self._lsts[i]) > 0:
```

```
                return (i, 0)
            i += 1

        return None

    def next(self):
        coords = self._next_coords(self.i, self.j)
        if coords is None:
            raise Exception("No more elements")
        else:
            self.i, self.j = coords
            return self._lsts[self.i][self.j]

    def has_next(self):
        coords = self._next_coords(self.i, self.j)
        return coords is not None
```