

# [CS 11] Prac 7m – 8 Neighbors In Grid

## Problem Statement

Consider an  $r \times c$  grid. We number the rows 0 to  $r - 1$  from top to bottom and the columns 0 to  $c - 1$  from left to right. We denote by  $(i, j)$  the cell at row  $i$  and column  $j$ .

Given a cell  $(i, j)$ , output all of its neighboring cells inside the grid, starting from the one north of it, and then going clockwise (so the next is northeast, then east etc.).

The neighbors of the cell marked `#` are marked `*` below:

```
. . . . .
. . . . .
. * * * .
. . * # * .
. . * * * .
. . . . .
. . . . .
```

Copy

Formally, two cells are neighbors in the above sense iff the  $\infty$ -norm distance between them is exactly 1.

## Task Details

Your task is to implement a function called `grid_neighs8`. This function has four `int` parameters  $r, c, i$  and  $j$ .

The function must return a *generator* that generates pairs of `int`s, as described in the problem statement.

Note that your generator must be **as lazy as possible**. It should yield each resulting next element as soon as it has enough information, and it should produce these results while advancing the input generators for as little as possible.

## Restrictions

(See 7a for more restrictions)

For this problem:

- Loops and lists are allowed.
- Up to 8 function definitions are allowed.
- Recursion is **disallowed**. (The recursion limit has been greatly reduced.)
- Sets and dictionaries are allowed.
- Generators and comprehensions are allowed.
- The source code limit is 400.

## Example Calls

### Example 1 Function Call

```
[*grid_neighs8(7, 5, 5, 2)]
```

Copy

### Example 1 Return Value

```
[(4, 2), (4, 3), (5, 3), (6, 3), (6, 2), (6, 1), (5, 1), (4,
```

Copy

### Example 2 Function Call

```
[*grid_neighs8(6, 3, 5, 2)]
```

Copy

### Example 2 Return Value

```
[(4, 2), (5, 1), (4, 1)]
```


Copy

## Constraints

When your program is run:

- The function `grid_neighs8` will be called at most 200 times.
- $1 \leq r, c \leq 10^{10}$
- $1 \leq i \leq r$
- $1 \leq j \leq c$

## Scoring

- You get 80  points if you solve all test cases.


## Clarifications


No clarifications have been made at this time.


Report an issue


Submit solution


[CS 11]


Practice 7 


My submissions 


 **Points:** 80 (partial)

 **Time limit:** 6.0s

 **Memory limit:** 1G

 **Author:**  
kvatienza (Kevin Atienza)

 **Problem type**

 **Allowed languages**  
NONE, py3