YEAH: Assignment 4

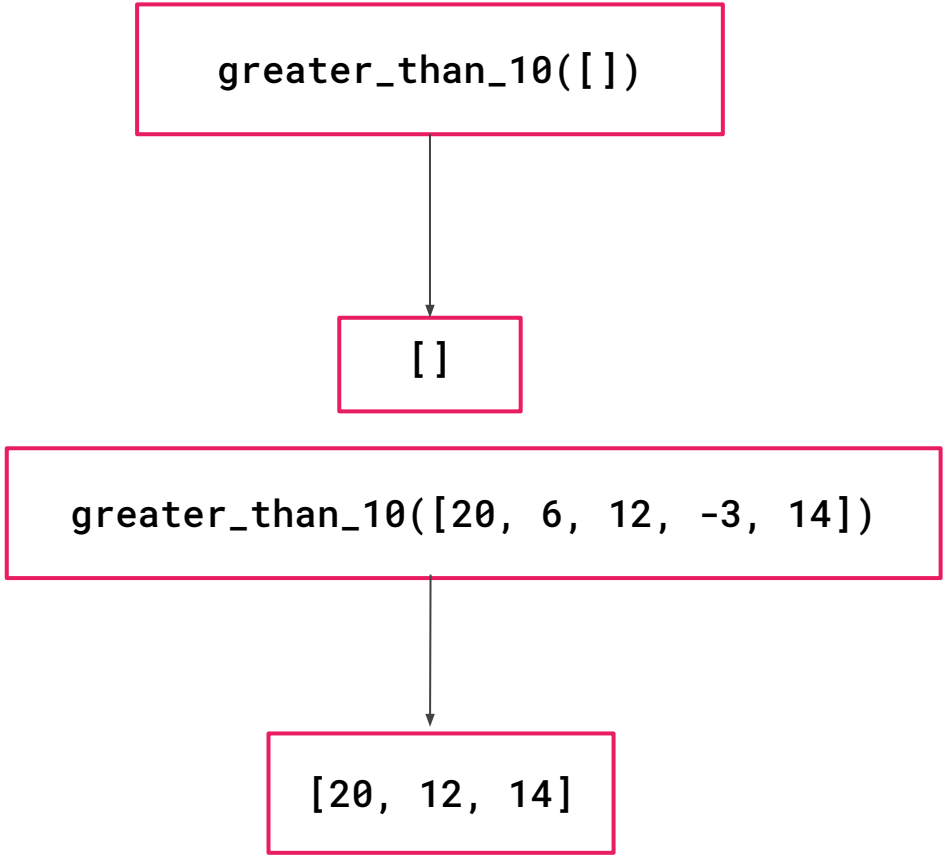
Tori and Kara

Part 1: Lists

Greater than 10

Goal: Given a list, return a new list of all the elements of the original list that were greater than 10

- If there are no elements greater than 10 or no elements at all, return an empty list
- Create a new list and don't modify the old list



Removing Duplicates from User Input

Goal: Prompt a user for integers until 0 is entered and return a list of the unique integers given

- `read_list()`
- `remove_duplicates(num_list)`

```
Enter value (0 to stop): 2
Enter value (0 to stop): 2
Enter value (0 to stop): 4
Enter value (0 to stop): 2
Enter value (0 to stop): 0
```

↓

```
[2, 4]
```

Hint: Check out this week's section problems if you're stuck
Hint: Think about types

Ziplists

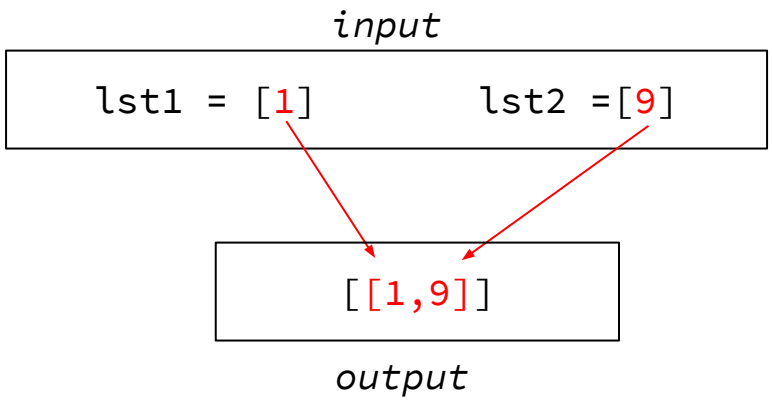
Goal: given two lists, pair up numbers (e.g. first element from list1 with first element of list2) and return a list of lists where each sublist is the pairing

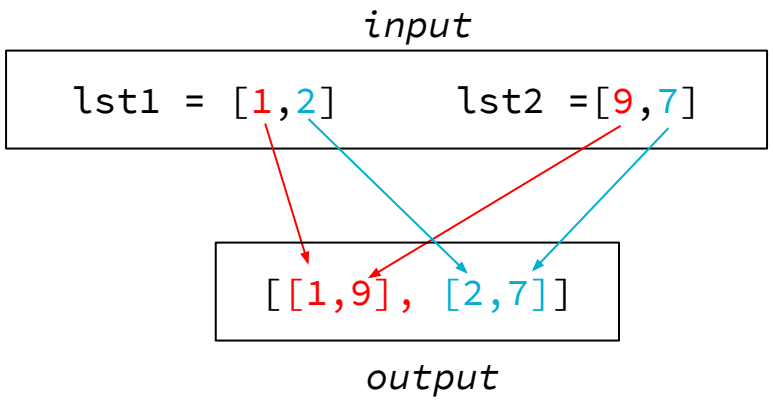
- Step 1: Make one pairing into a list
- Step 2: Add that list to a list of lists

If there are no elements in list1 and list2, just return an empty list

Hint: Think of how the two sides of a zipper come together

Ziplist Example





Part 2: Sand

[demo of what Sand should look like]

The Sand World

World

'r'	's'	'r'
'r'		'r'

```
[['r', 's', 'r'],
['r', None, 'r']]
```

'r'		'r'
'r'	's'	'r'

```
[['r', None, 'r'],
['r', 's', 'r']]
```

Elements

Key	
Sand	's'
Rock	'r'
Empty	None

REMEMBER

A grid's x,y coordinates are reversed

Element at x=1, y=0 is at grid[0][1]

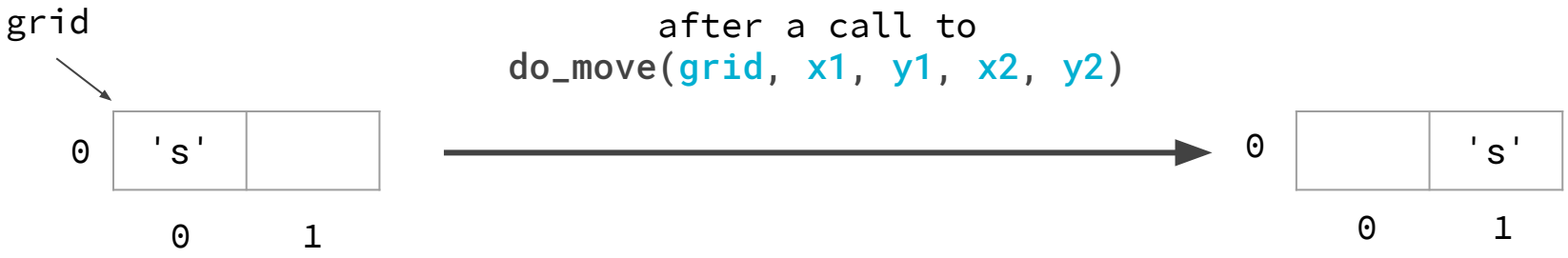
For this presentation, when we use (x,y) we are referring to traditional coordinates, not grid coordinates

---

Milestone 1: Moving elements in grid

```
do_move(grid, x1, y1, x2, y2)
```

Take what is at (x1,y1) of the grid, move it to (x2,y2), and set (x1,y1) to be None



Milestone 2: Checking legal moves

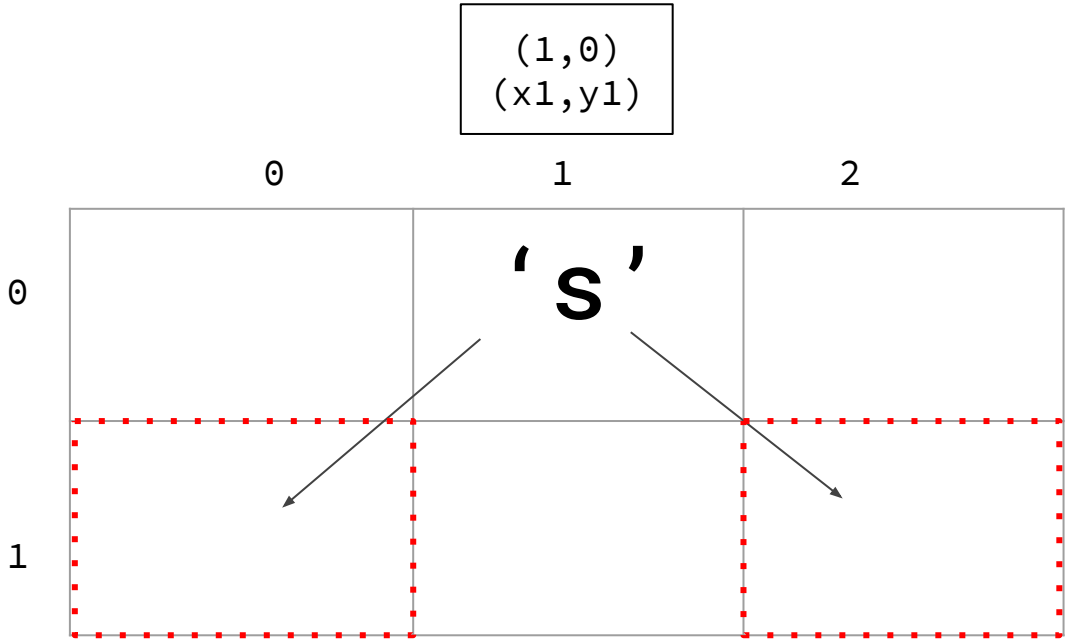
There are 3 legal moves:

- moving straight down, if (x2,y2) has nothing in it
- moving diagonally left, if the spot immediately to left of (x1,y1) is empty and if (x2,y2) has nothing in it
- moving diagonally right, if the spot immediately to right of (x1,y1) is empty and if (x2,y2) has nothing in it

For all of these moves, the destination (x2,y2) must be within the grid boundaries

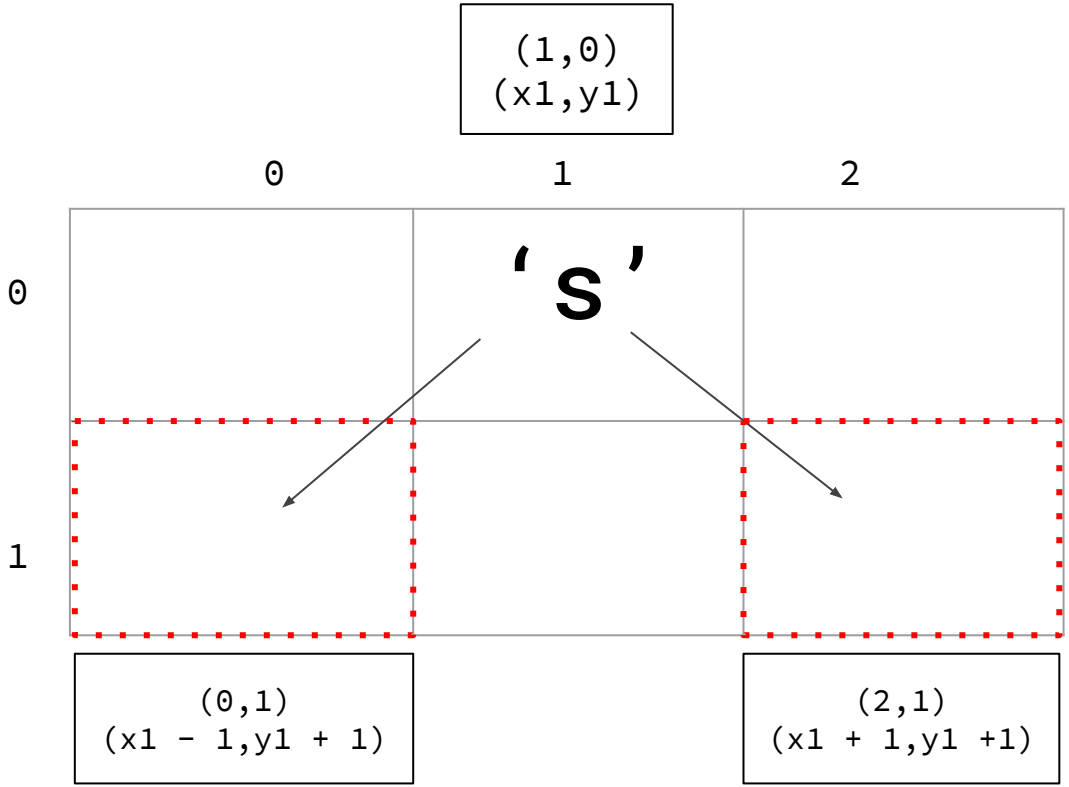
Diagonal Moves

What are the coordinates of destinations for diagonal moves?



Diagonal Moves

What are the coordinates of destinations for diagonal moves?



Milestone 3: Gravity

- Step 1: check if straight down is valid
- Step 2: if straight down didn't work, check if you can go diagonal
- Step 3: if no legal moves, don't do anything

before

'r'	's'	'r'
'r'		'r'

after

'r'		'r'
'r'	's'	'r'

sand at (1, 0) moved directly down to (1,1)

before

	's'	'r'
	'r'	'r'

after

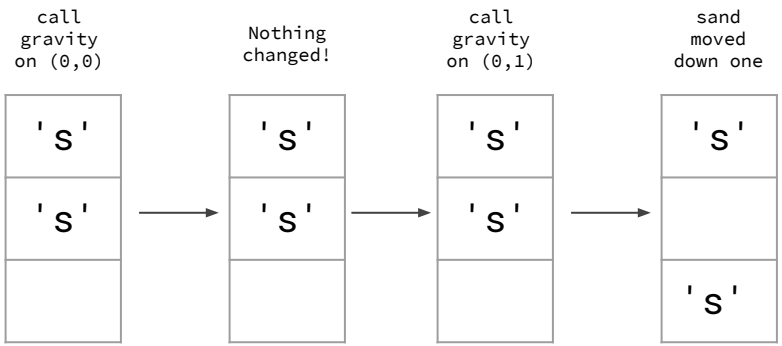
		'r'
's'	'r'	'r'

sand at (1, 0) moved diagonally left to (0,1)

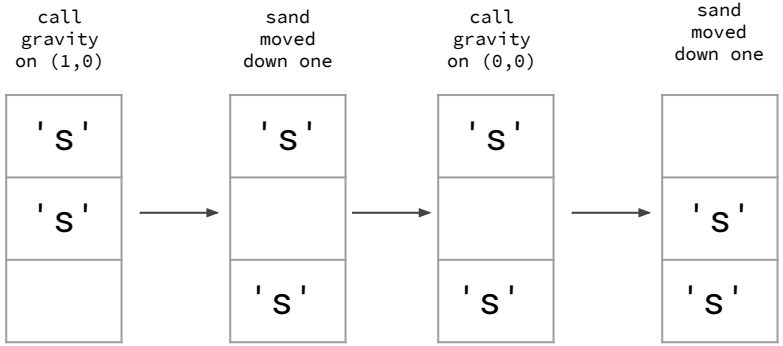
Milestone 4: Loop through the whole grid

- For each (x,y) location, call gravity
- But what order should we iterate over all these locations?
  - ORDER MATTERS

Iterating from top to bottom



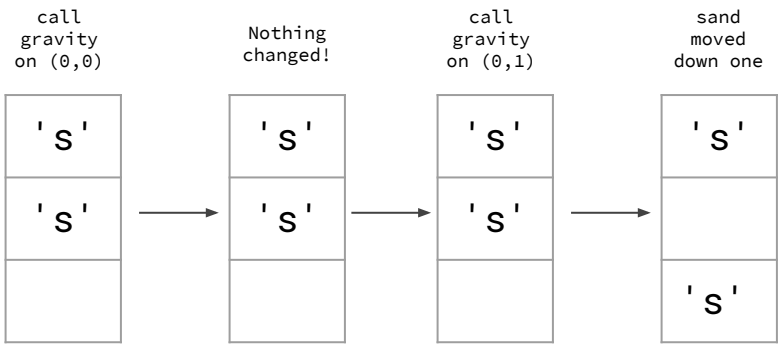
Iterating from bottom to top



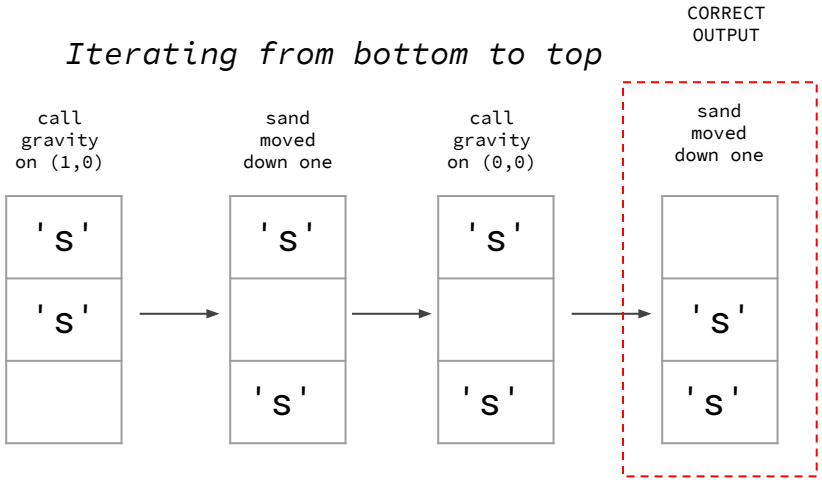
Milestone 4: Loop through the whole grid

- For each (x,y) location, call gravity
- But what order should we iterate over all these locations?
  - ORDER MATTERS

Iterating from top to bottom



Iterating from bottom to top

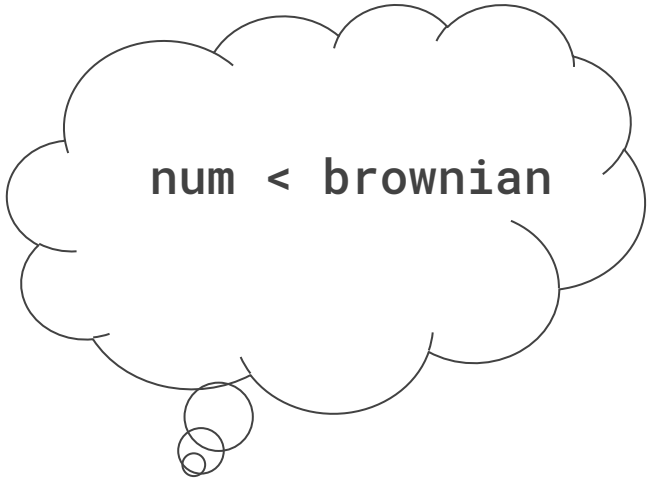


Milestone 5: Create Brownian motion

Giving sand Brownian motion:

```
num = random.randrange(100)
```

```
coin = random.randrange(2)
```



Make sure your functions work in harmony!

- All of your functions should work in harmony with one another
- If you already validate values (error checking) in one function, don't have to do it again in some helper function

Good luck!