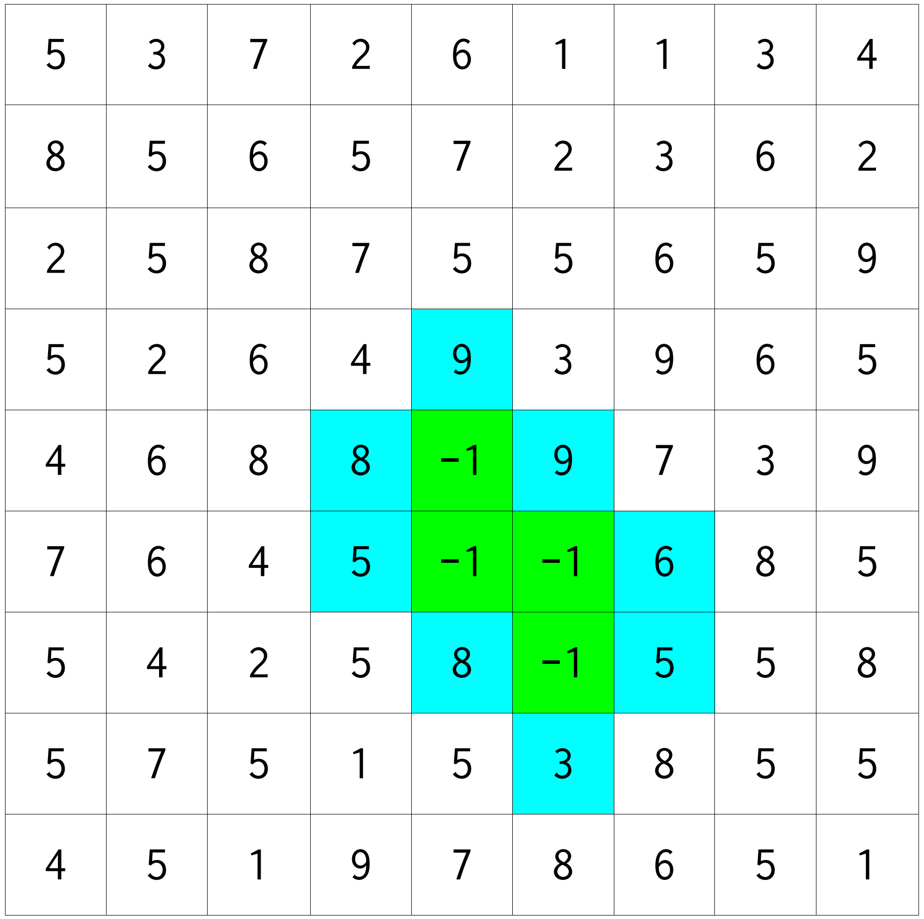
Program Design

Invasion Percolation: Neighbors

5	3	7	2	6	1	1	3	4
8	5	6	5	7	2	3	6	2
2	5	8	7	5	5	6	5	9
5	2	6	4	9	3	9	6	5
4	6	8	8	🟩	9	7	3	9
7	6	4	5	🟩	🟩	6	8	5
5	4	2	5	8	🟩	5	5	8
5	7	5	1	5	3	8	5	5
4	5	1	9	7	8	6	5	1

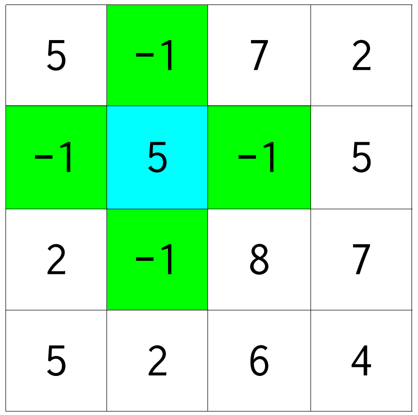
Need to find neighbors of marked cells



5	3	7	2	6	1	1	3	4
8	5	6	5	7	2	3	6	2
2	5	8	7	5	5	6	5	9
5	2	6	4	9	3	9	6	5
4	6	8	8	-1	9	7	3	9
7	6	4	5	-1	-1	6	8	5
5	4	2	5	8	-1	5	5	8
5	7	5	1	5	3	8	5	5
4	5	1	9	7	8	6	5	1

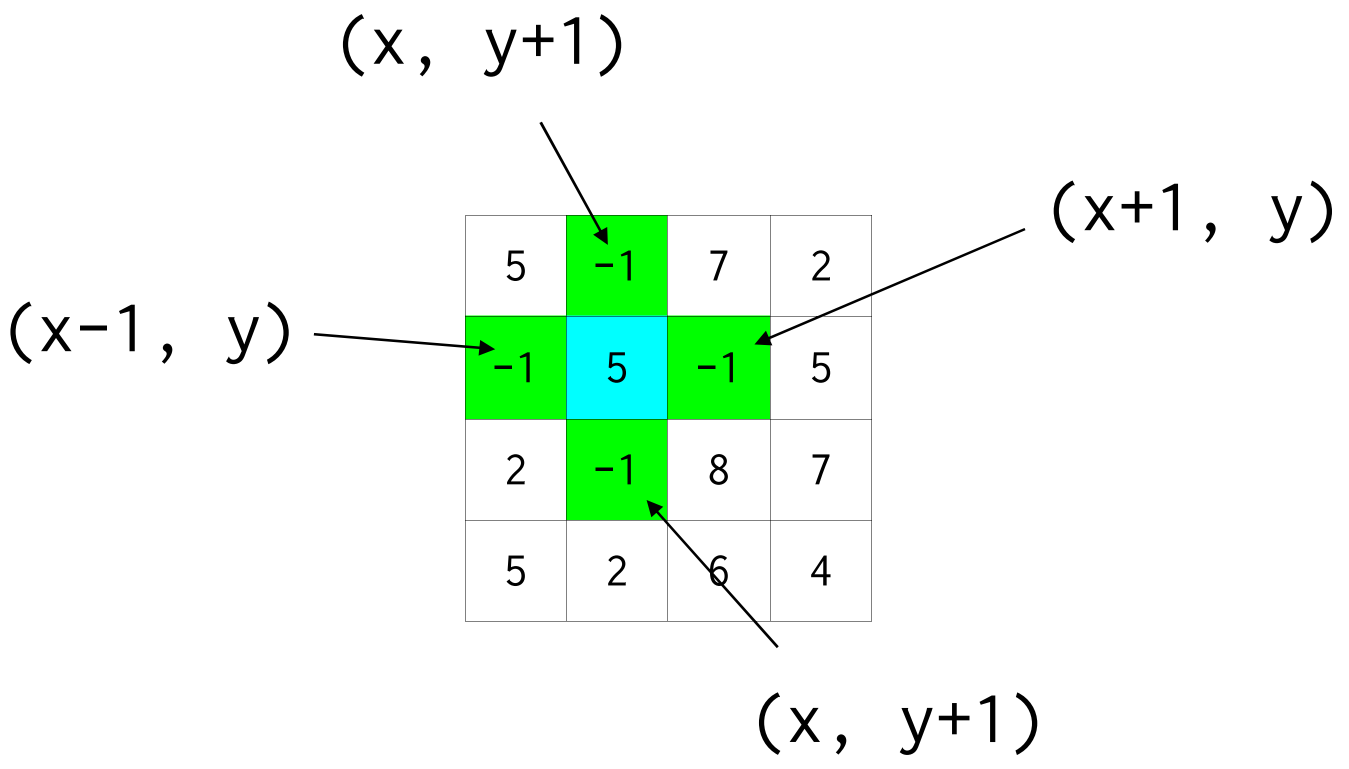
Need to find neighbors of marked cells

...which are marked with -1



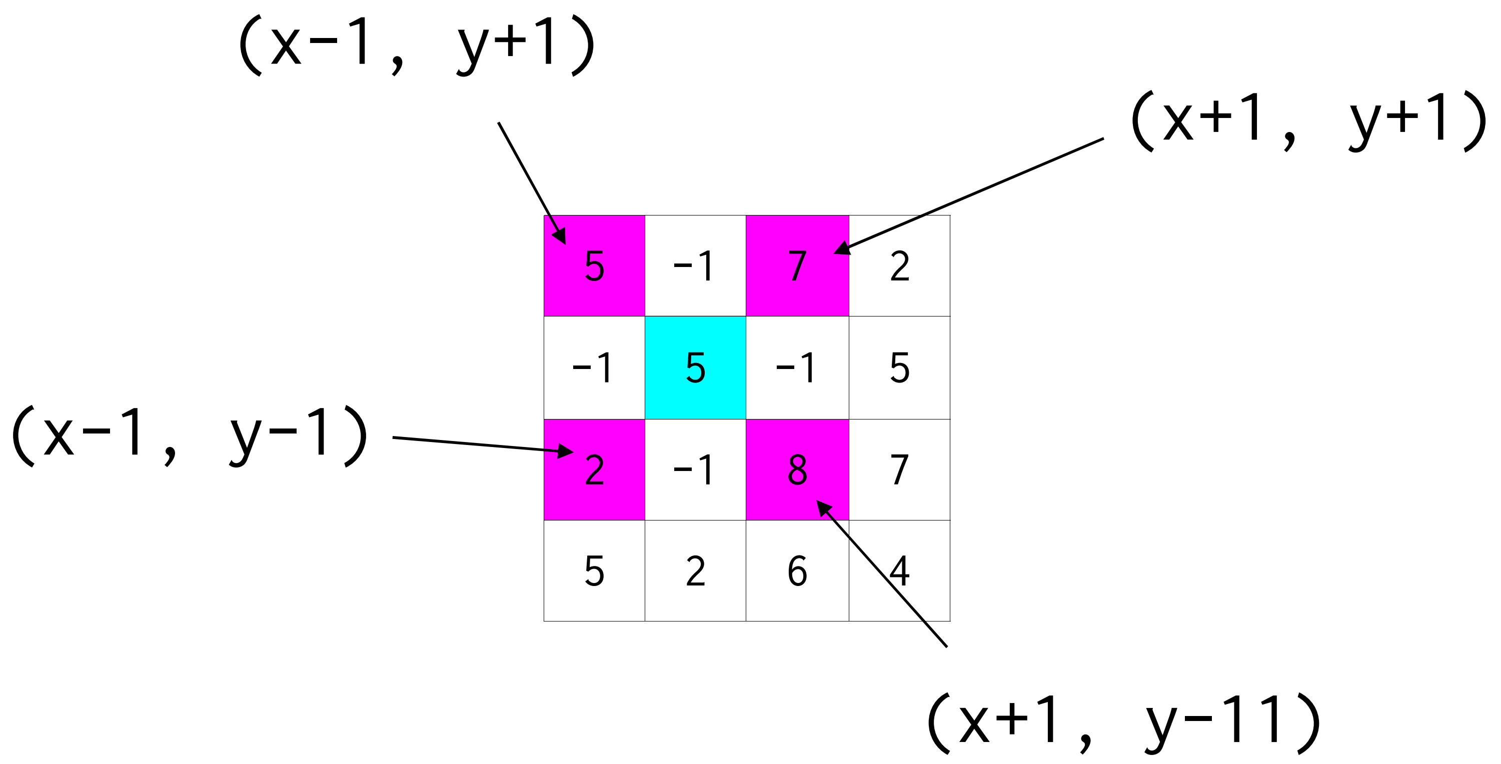
5	−1	7	2
−1	5	−1	5
2	−1	8	7
5	2	6	4

Blue cell is a neighbor if any of the green cells have already been filled

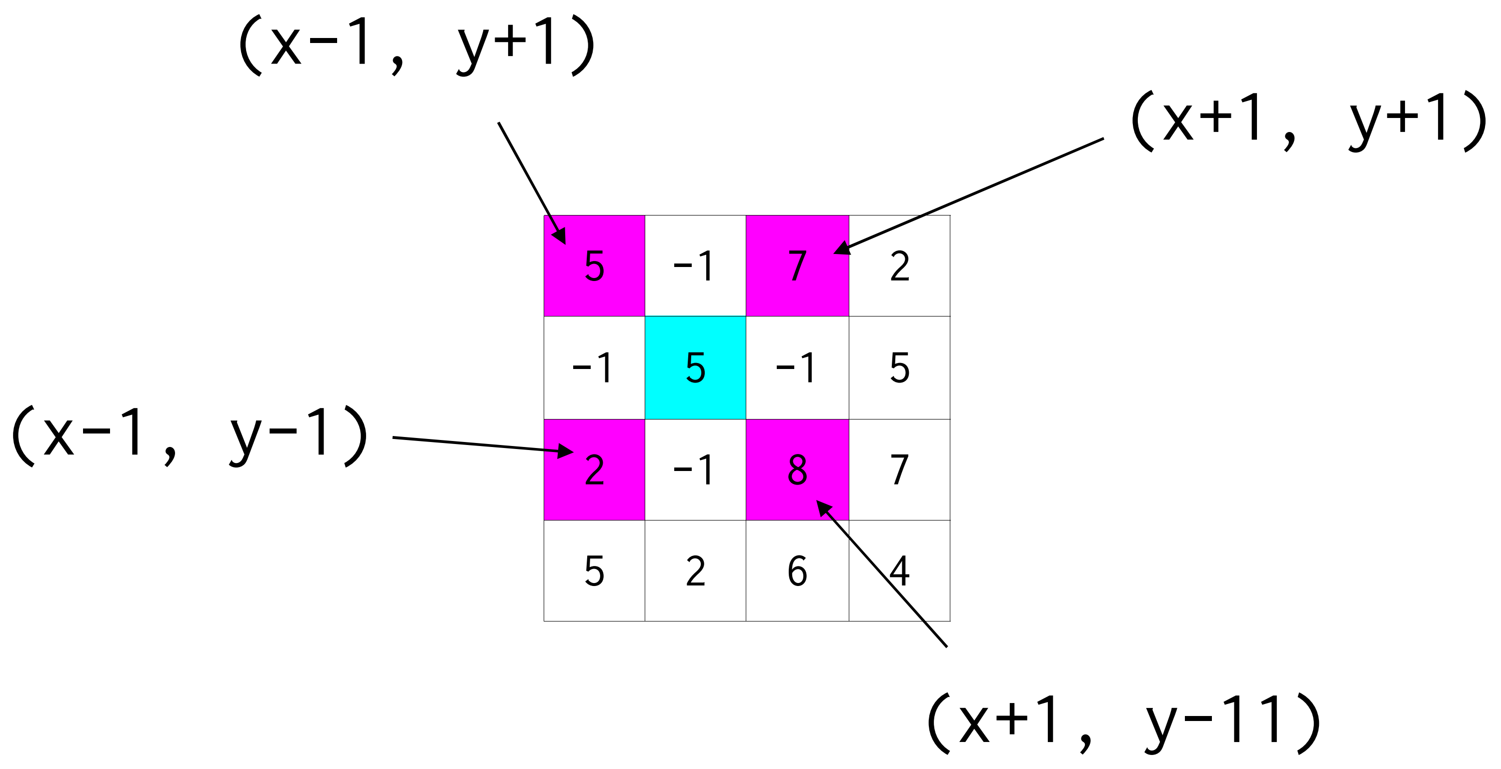


Those cells coordinates are the center cell's

±1 to either x or y



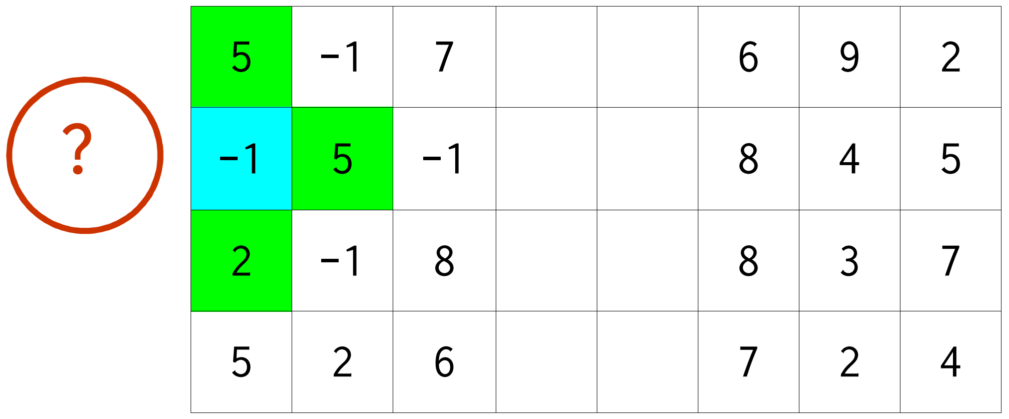
We're assuming diagonal cells don't count



We're assuming diagonal cells don't count

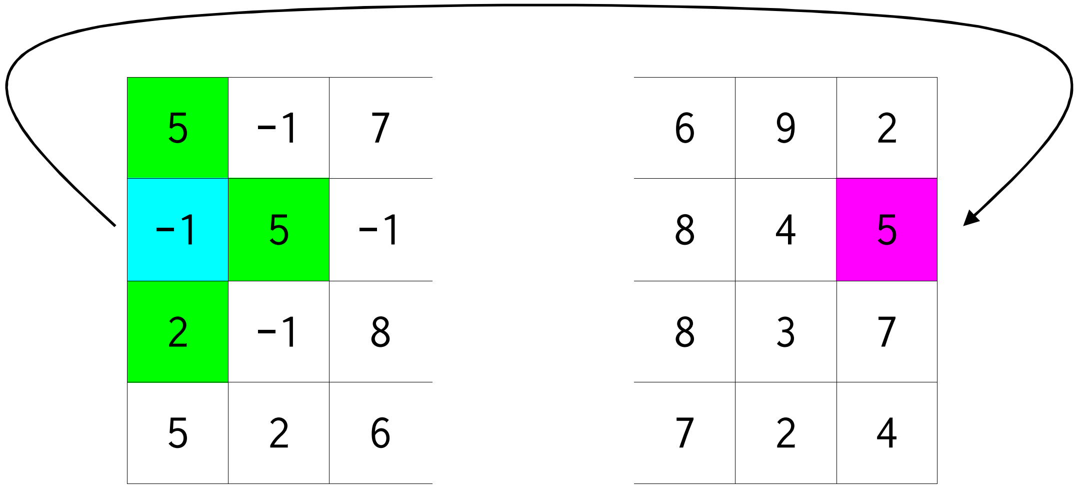
Need to check the science...

```
# Is a cell a candidate for filling?
# Version 1: has bugs!
for x in range(N):
  for y in range(N):
    if is_filled(grid, x-1, y) \
    or is_filled(grid, x+1, y) \
    or is_filled(grid, x, y-1) \
    or is_filled(grid, x, y+1):
      ...cell (x, y) is a candidate...
```

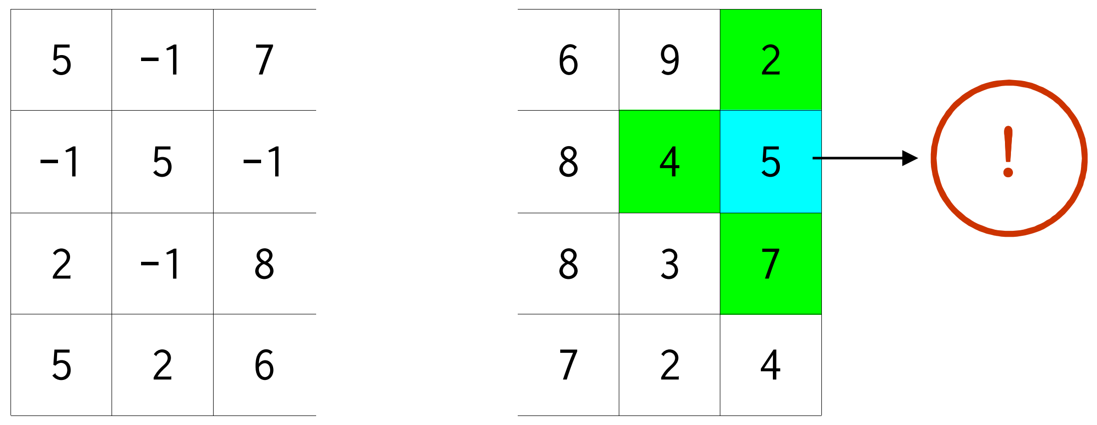


5	-1	7			6	9	2
-1	5	-1			8	4	5
2	-1	8			8	3	7
5	2	6			7	2	4

What do we do at the edges?



[0-1] == -1 wraps around to the other edge



[(N-1)+1] == N is an out-of-bounds exception

```
# Is a cell a candidate for filling?
# Version 2: long-winded
for x in range(N):
  for y in range(N):
    if x > 0:
      if is_filled(grid, x-1, y):
        ...cell (x, y) is a candidate...
    ...repeat for the other three cases...
```

```
# Is a cell a candidate for filling?
# Version 2: long-winded
for x in range(N):
  for y in range(N):
    if x > 0:
      if is_filled(grid, x-1, y):
        ...cell (x, y) is a candidate...
    ...repeat for the other three cases...
```

"Code that is repeated in two or more places will eventually be wrong in at least one."

```
# Is a cell a candidate for filling?
# Version 3: good enough for production
for x in range(N):
  for y in range(N):
    if (x > 0) and is_filled(x-1, y)\
    or (x < N-1) and is_filled(x+1, y)\
    or (y > 0) and is_filled(x, y-1)\
    or (y < N-1) and is_filled(x, y+1):
      ...cell (x, y) is a candidate...
```

```
if (x > 0) and is_filled(x-1, y)
```

Not on the

left edge

```
if (x > 0) and is_filled(x-1, y)
```

Not on the left edge

Neighbor is already filled

Short-circuit evaluation

```
if (x > 0) and is_filled(x-1, y)
```

Not on the left edge

Neighbor is already filled

```
if sanity_check and some_other_test:
```

Make sure the second test won't blow up

Only execute if it's safe to do so

Don't try second part if first part is False because the answer is already known created by

Greg Wilson

May 2010