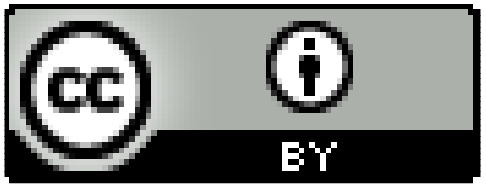
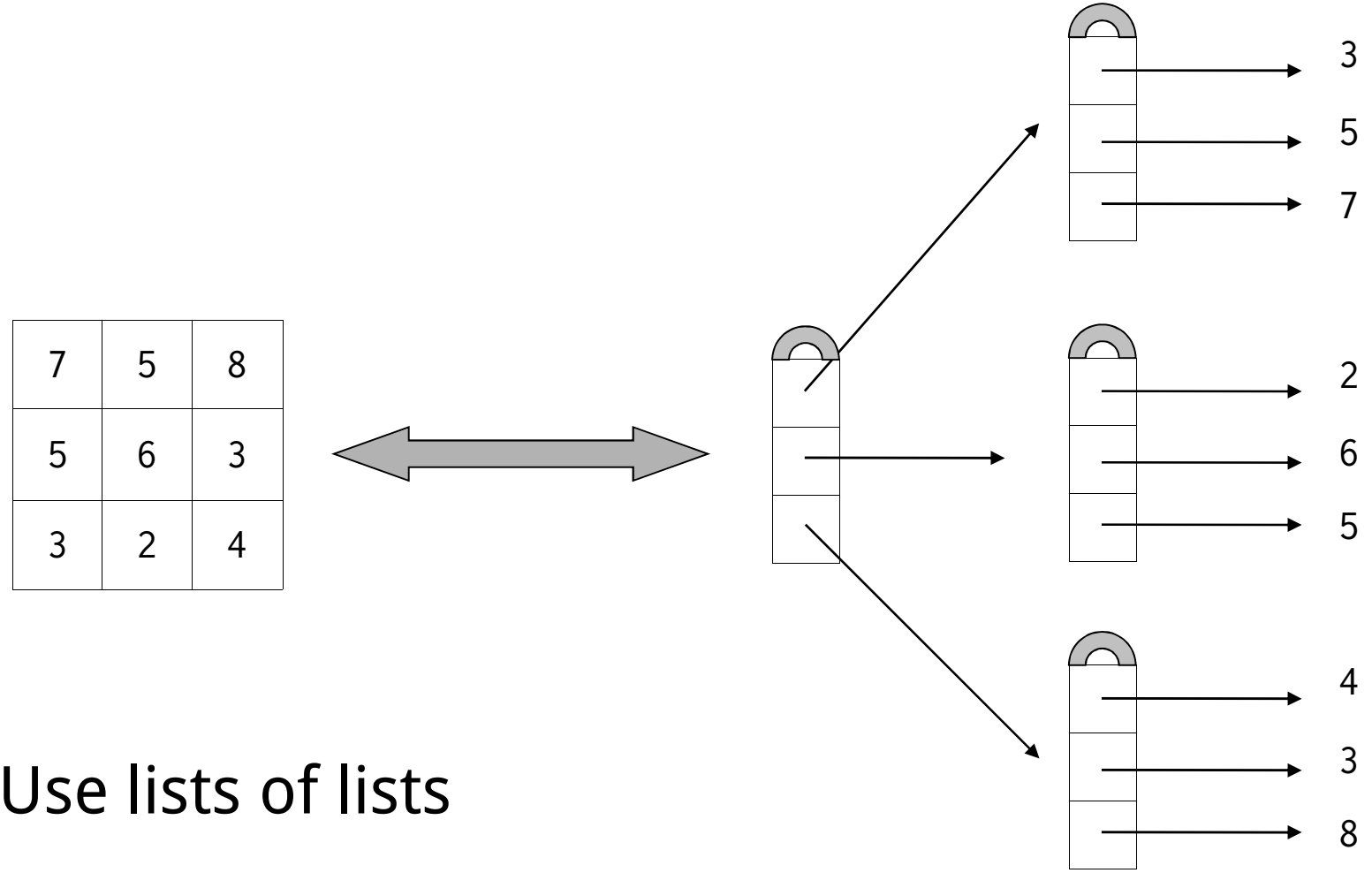
Program Design

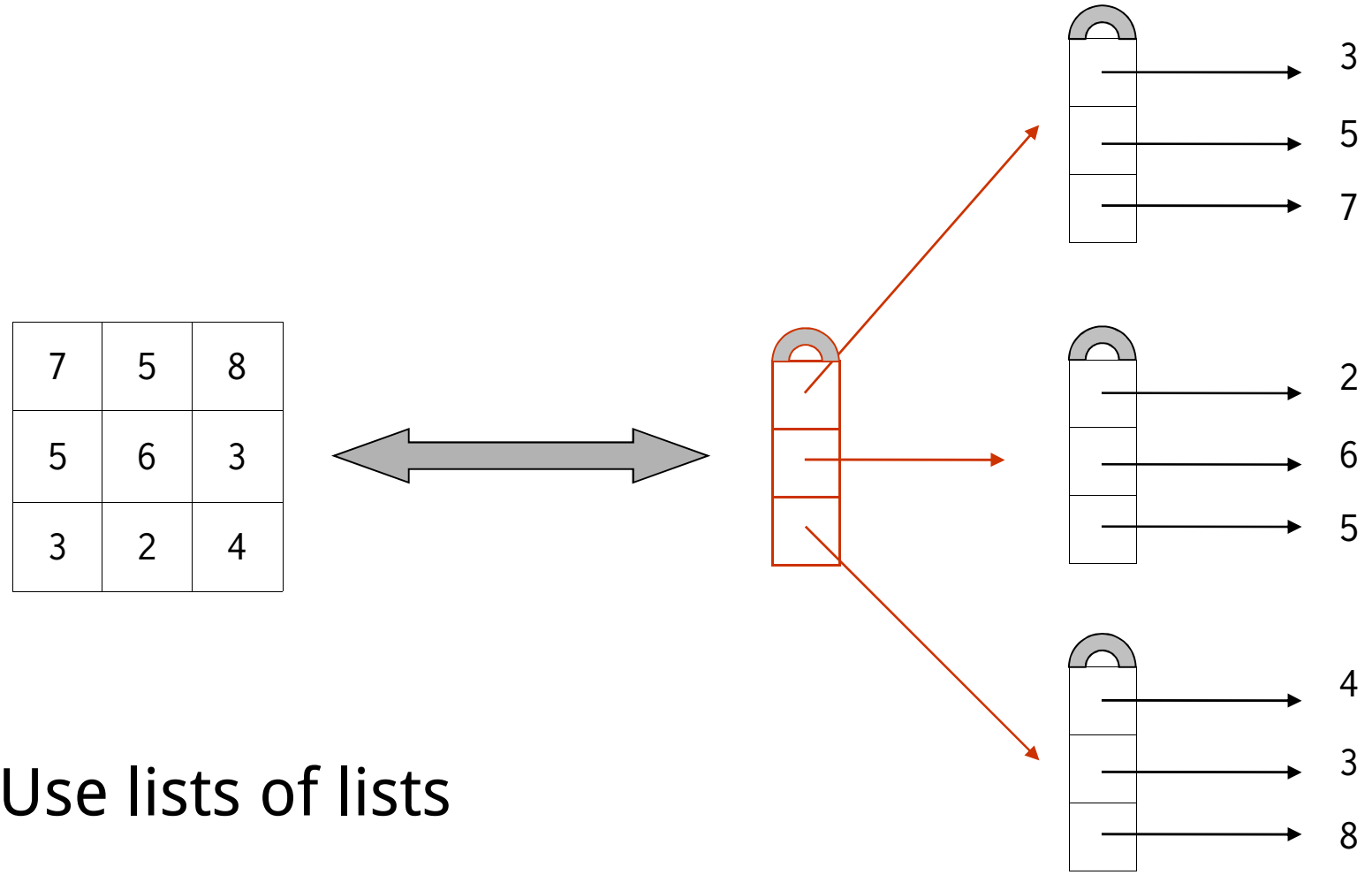
Invasion Percolation: Aliasing





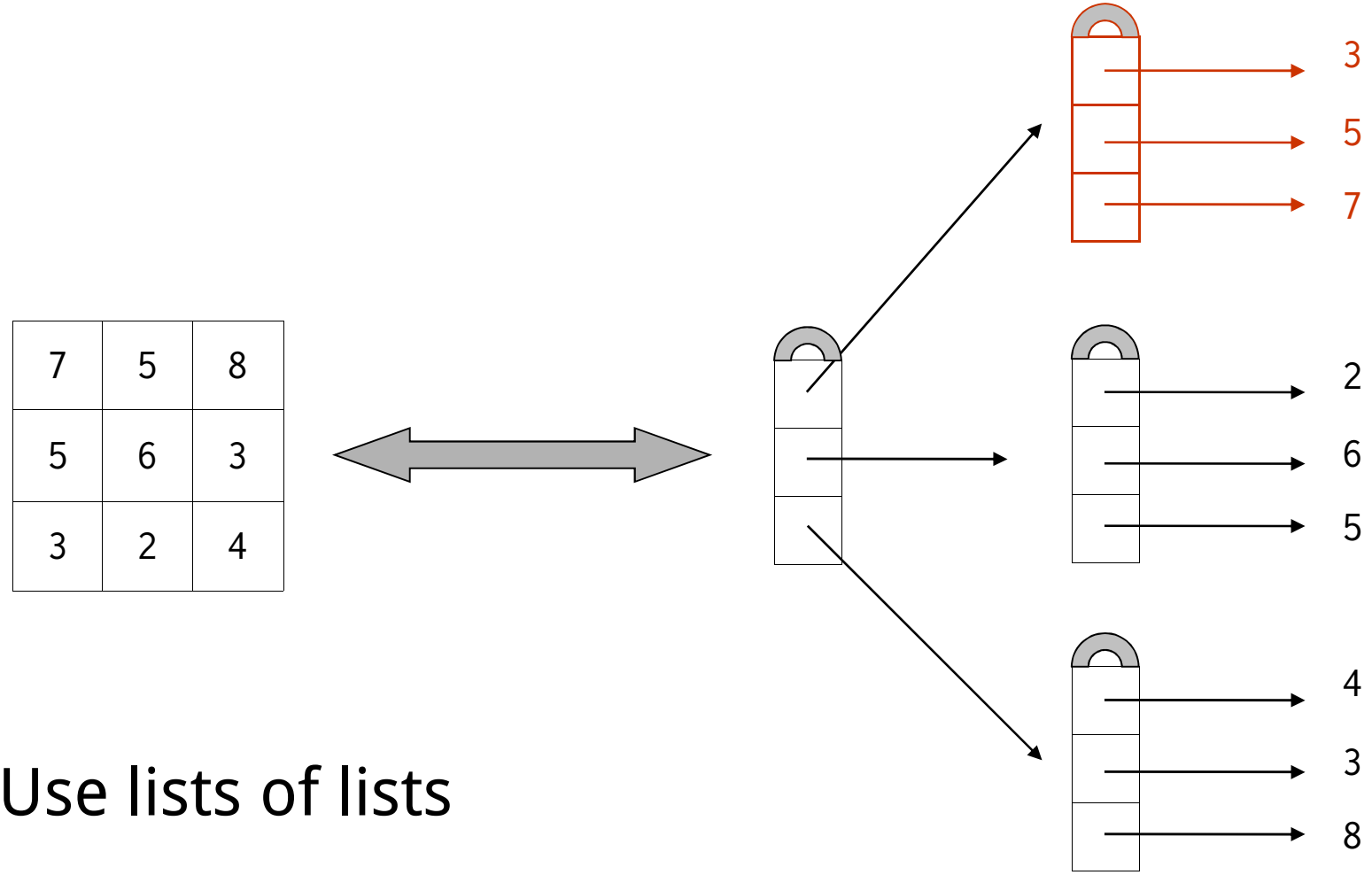
Use lists of lists

to implement a 2D grid



Use lists of lists

to implement a 2D grid



Use lists of lists

to implement a 2D grid

```
# Correct code
assert N > 0, "Grid size must be positive"
assert N%2 == 1, "Grid size must be odd"
grid = []
for x in range(N):
  grid.append([])
  for y in range(N):
    grid[-1].append(1)
```

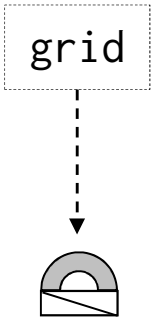
```
# Incorrect code
assert N > 0, "Grid size must be positive"
assert N%2 == 1, "Grid size must be odd"
grid = []
EMPTY = []
for x in range(N):
  grid.append(EMPTY)
  for y in range(N):
    grid[-1].append(1)
```

```
# Incorrect code
assert N > 0, "Grid size must be positive"
assert N%2 == 1, "Grid size must be odd"
grid = []
EMPTY = []
for x in range(N):
  grid.append(EMPTY)
  for y in range(N):
    grid[-1].append(1)
```

```
# Incorrect code
assert N > 0, "Grid size must be positive"
assert N%2 == 1, "Grid size must be odd"
grid = []
EMPTY = []
for x in range(N):
  grid.append(EMPTY)
  for y in range(N):
    grid[-1].append(1)
```

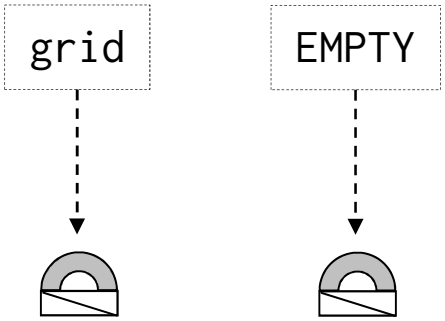
"Aren't meaningful variable names supposed to be a good thing?"

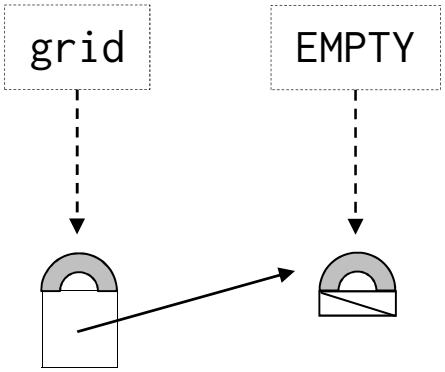
```
grid = []
```



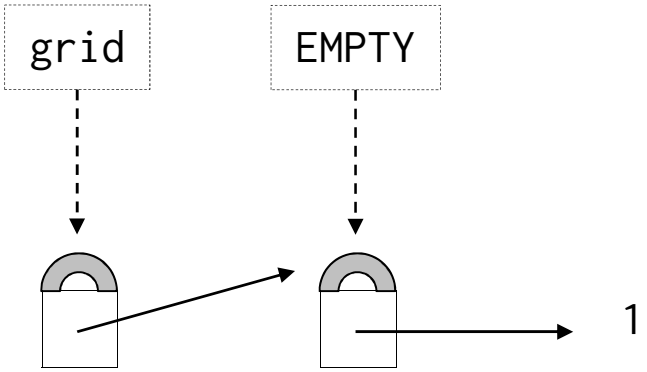
```
grid = []
```

```
EMPTY = []
```

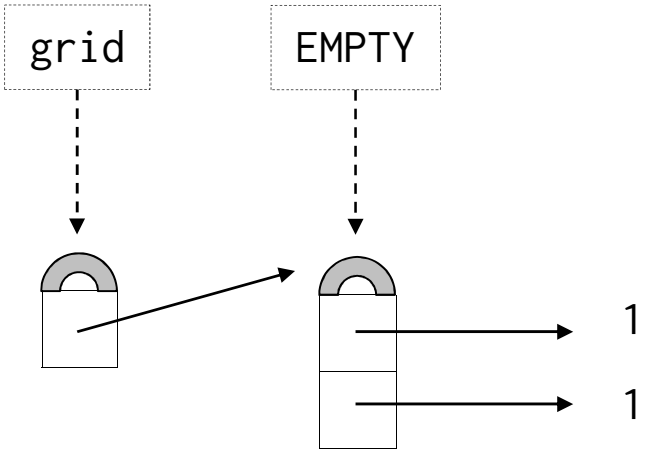




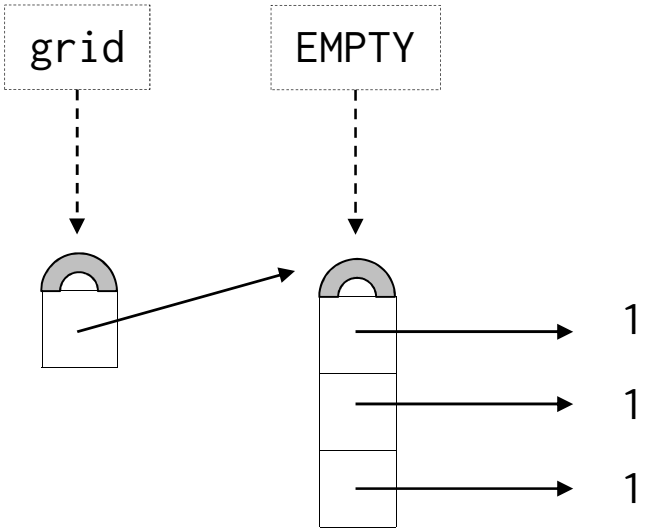

```
grid = []
EMPTY = []
for x in range(N): # x == 0
  grid.append(EMPTY)
```



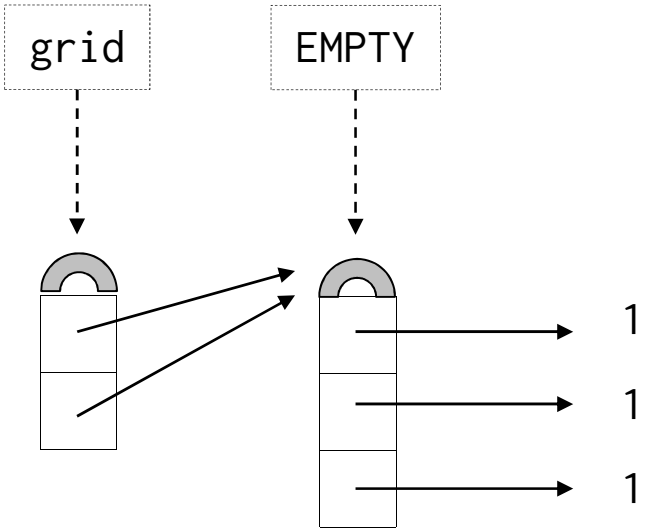

```
grid = []
EMPTY = []
for x in range(N): # x == 0
  grid.append(EMPTY)
  for y in range(N): # y == 0
    grid[-1].append(1)
```



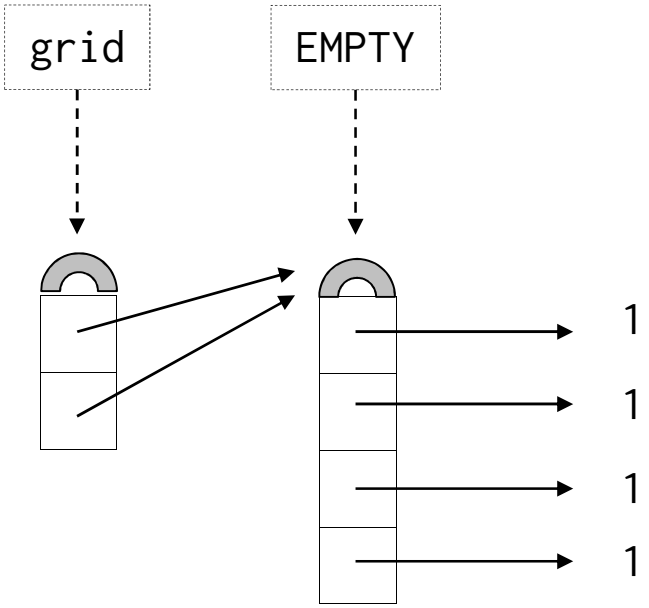

```
grid = []
EMPTY = []
for x in range(N): # x == 0
  grid.append(EMPTY)
  for y in range(N): # y == 1
    grid[-1].append(1)
```



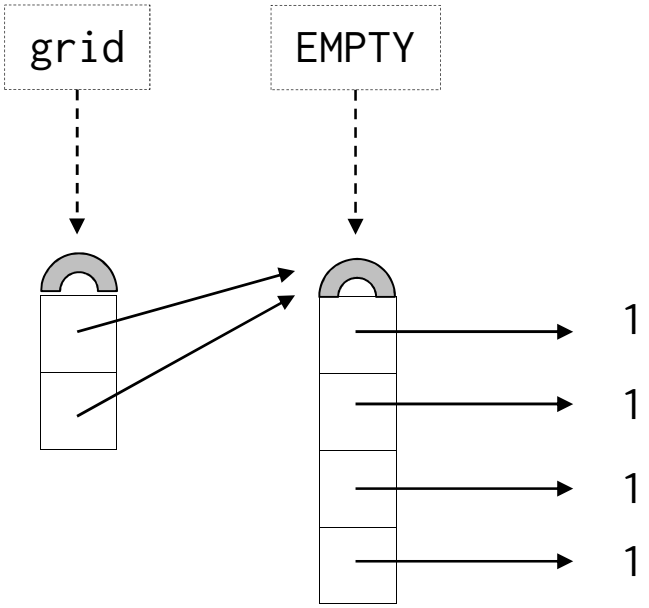

```
grid = []
EMPTY = []
for x in range(N): # x == 0
  grid.append(EMPTY)
  for y in range(N): # y == 2
    grid[-1].append(1)
```




```
grid = []
EMPTY = []
for x in range(N): # x == 1
  grid.append(EMPTY)
```




```
grid = []
EMPTY = []
for x in range(N): # x == 1
  grid.append(EMPTY)
  for y in range(N): # y == 0
    grid[-1].append(1)
```




```
grid = []
EMPTY = []
for x in range(N): # x == 1
  grid.append(EMPTY)
  for y in range(N): # y == 0
    grid[-1].append(1)
```

You see the problem...

Indirection allows aliasing

Indirection allows aliasing

Aliasing can be useful

Indirection allows aliasing

Aliasing can be useful

(In fact, sometimes it's indispensible)

Indirection allows aliasing

Aliasing can be useful

(In fact, sometimes it's indispensible)

But it's also a rich source of bugs

Indirection allows aliasing

Aliasing can be useful

(In fact, sometimes it's indispensible)

But it's also a rich source of bugs

When in doubt, draw a picture!

Indirection allows aliasing

Aliasing can be useful

(In fact, sometimes it's indispensible)

But it's also a rich source of bugs

When in doubt, draw a picture!

Tools that do this automatically exist...

Indirection allows aliasing

Aliasing can be useful

(In fact, sometimes it's indispensible)

But it's also a rich source of bugs

When in doubt, draw a picture!

Tools that do this automatically exist...

...but none has really taken off (yet)

created by

Greg Wilson

May 2010