

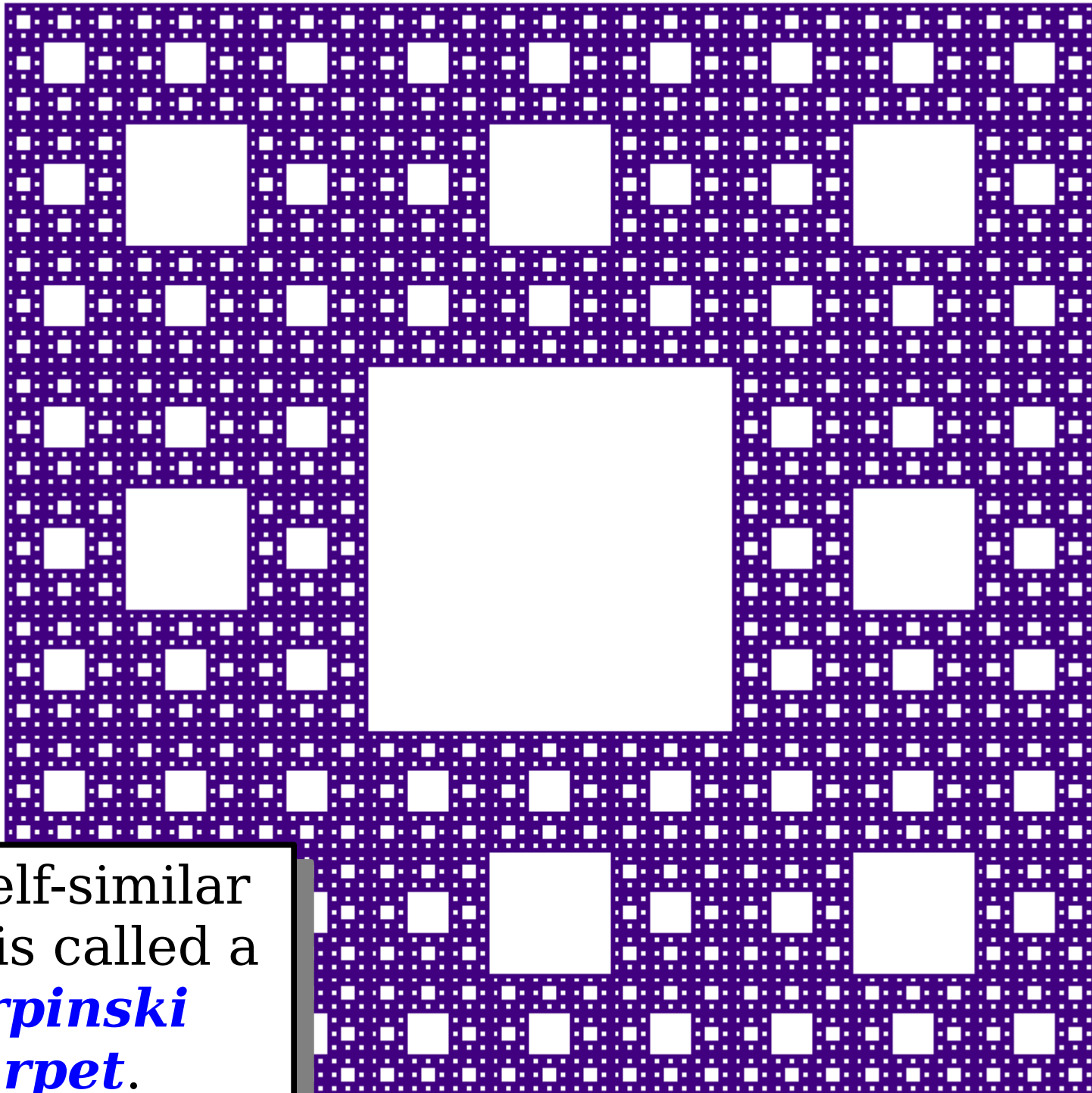
Thinking Recursively

Part III

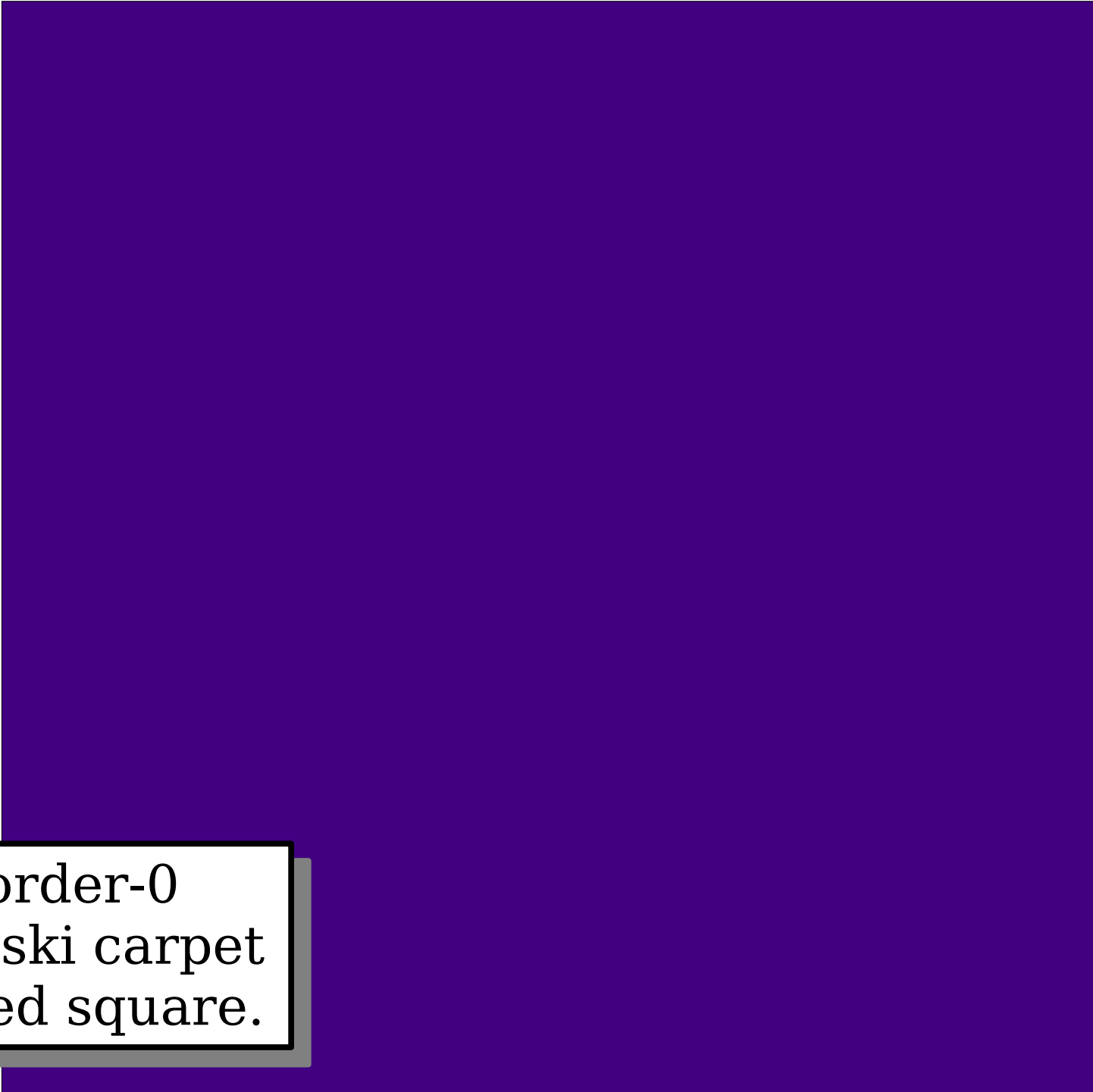
Outline for Today

- ***Iteration + Recursion***
 - Combining two techniques together.
- ***Enumerating Permutations***
 - What order should we do things?
- ***Enumeration, Generally***
 - How to think about enumeration problems.

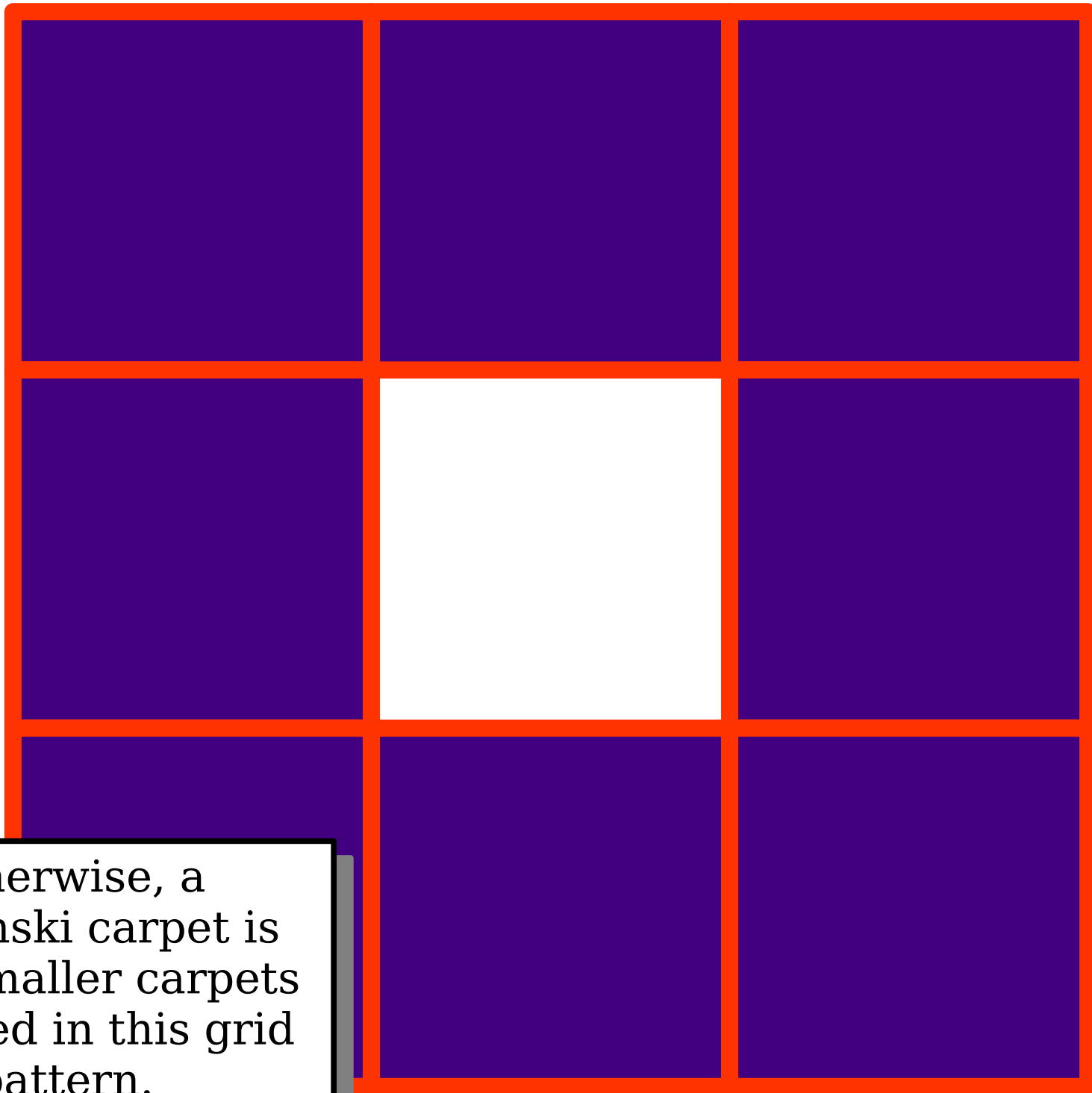
More On Self-Similarity



This self-similar shape is called a ***Sierpinski carpet.***



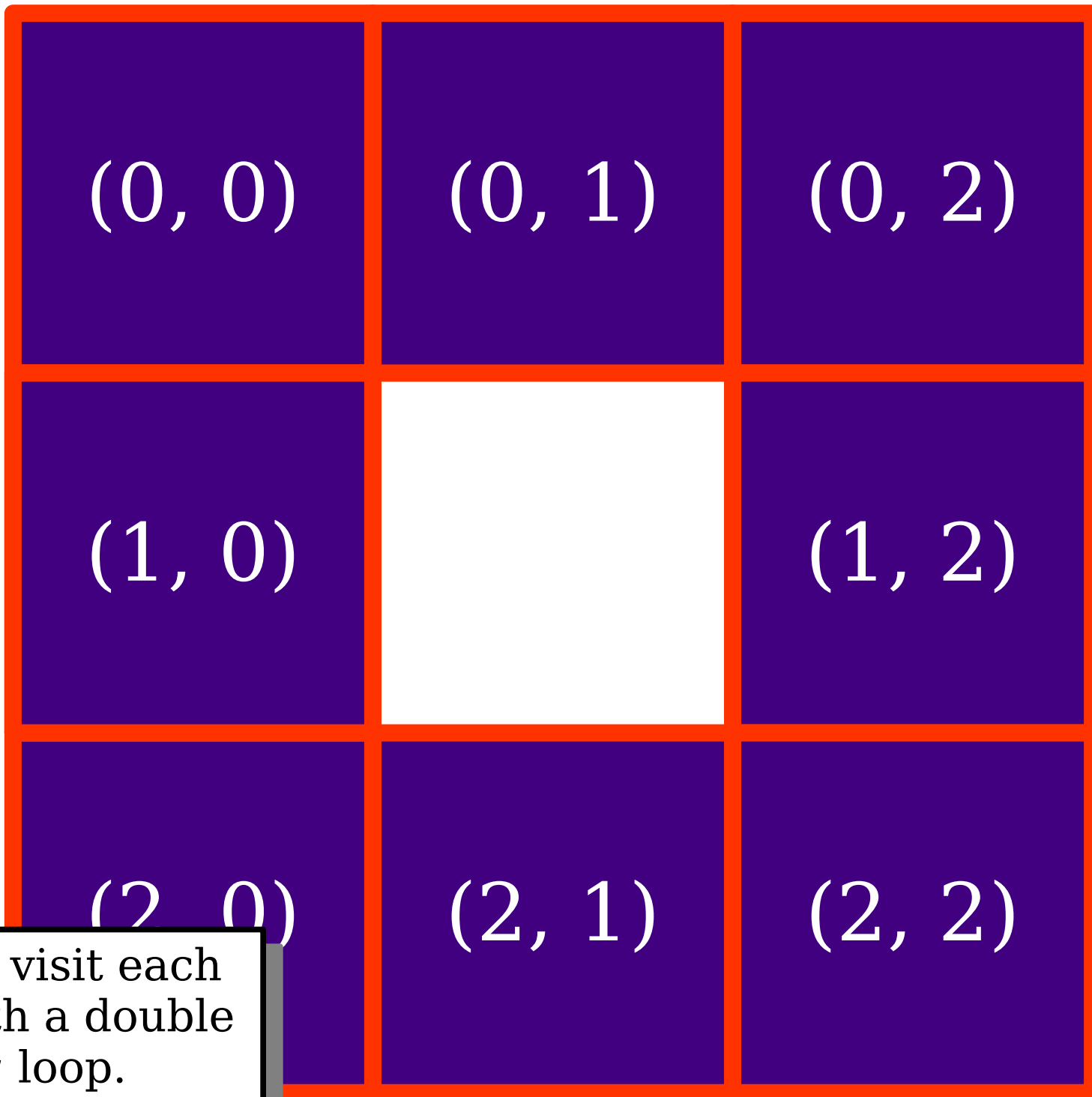
An order-0
Sierpinski carpet
is a filled square.



Otherwise, a Sierpinski carpet is eight smaller carpets arranged in this grid pattern.

$(0, 0)$	$(0, 1)$	$(0, 2)$
$(1, 0)$		$(1, 2)$
$(2, 0)$	$(2, 1)$	$(2, 2)$

Label each square
with its (row, col).



We can visit each spot with a double for loop.

Iteration + Recursion

- It's completely reasonable to mix iteration and recursion in the same function.
- Here, we're firing off eight recursive calls, and the easiest way to do that is with a double for loop.
- Recursion doesn't mean "the absence of iteration." It just means "solving a problem by solving smaller copies of that same problem."

Time-Out for Announcements!

Assignment 2

- Assignment 2 was due today at the start of lecture.
- The grace period for late submissions ends this Sunday at 11:30AM Pacific time.
- Have questions? Ask them on EdStem, or email your section leader!

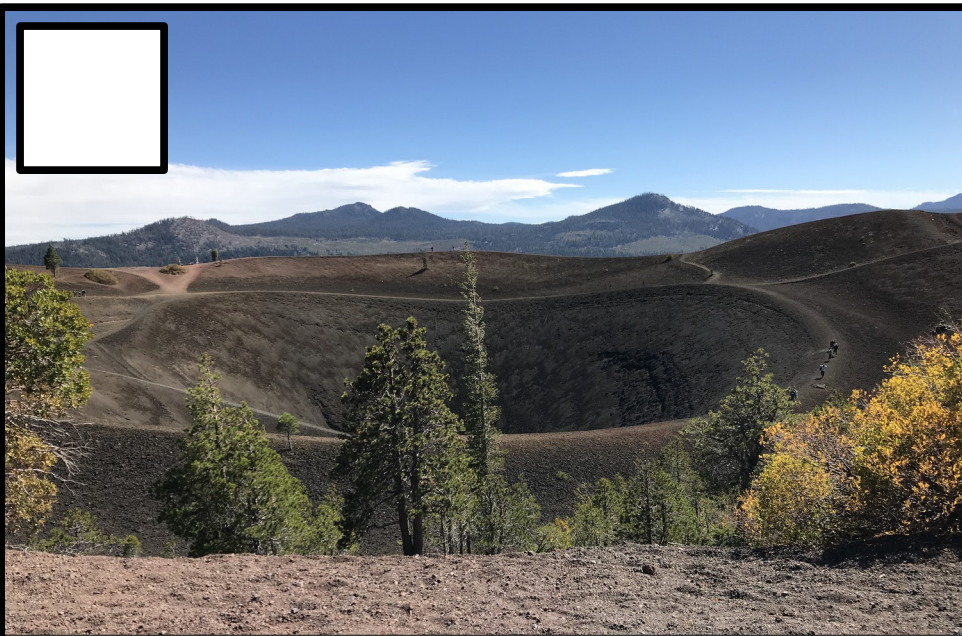
Assignment 3

- Assignment 3 (***Recursion!***) goes out today. It's due next Friday at the start of class.
 - Play around with recursion and recursive problem-solving!
- ***This assignment may be completed in pairs.***
Some reminders:
 - You are not required to work in a pair. It's totally fine to work independently.
 - If you do work in a pair, you must work with someone else in your discussion section.
 - ***Work together, not separately.*** Doing only half the assignment teaches you less than half the concepts. Working collaboratively and interactively with your partner will improve your learning outcomes.

(The Curtain Rises on Act II)

Enumerating Permutations

A ***permutation*** is a rearrangement of the elements of a sequence.



Lassen Volcanic National Park



Yosemite National Park



Coconino National Forest



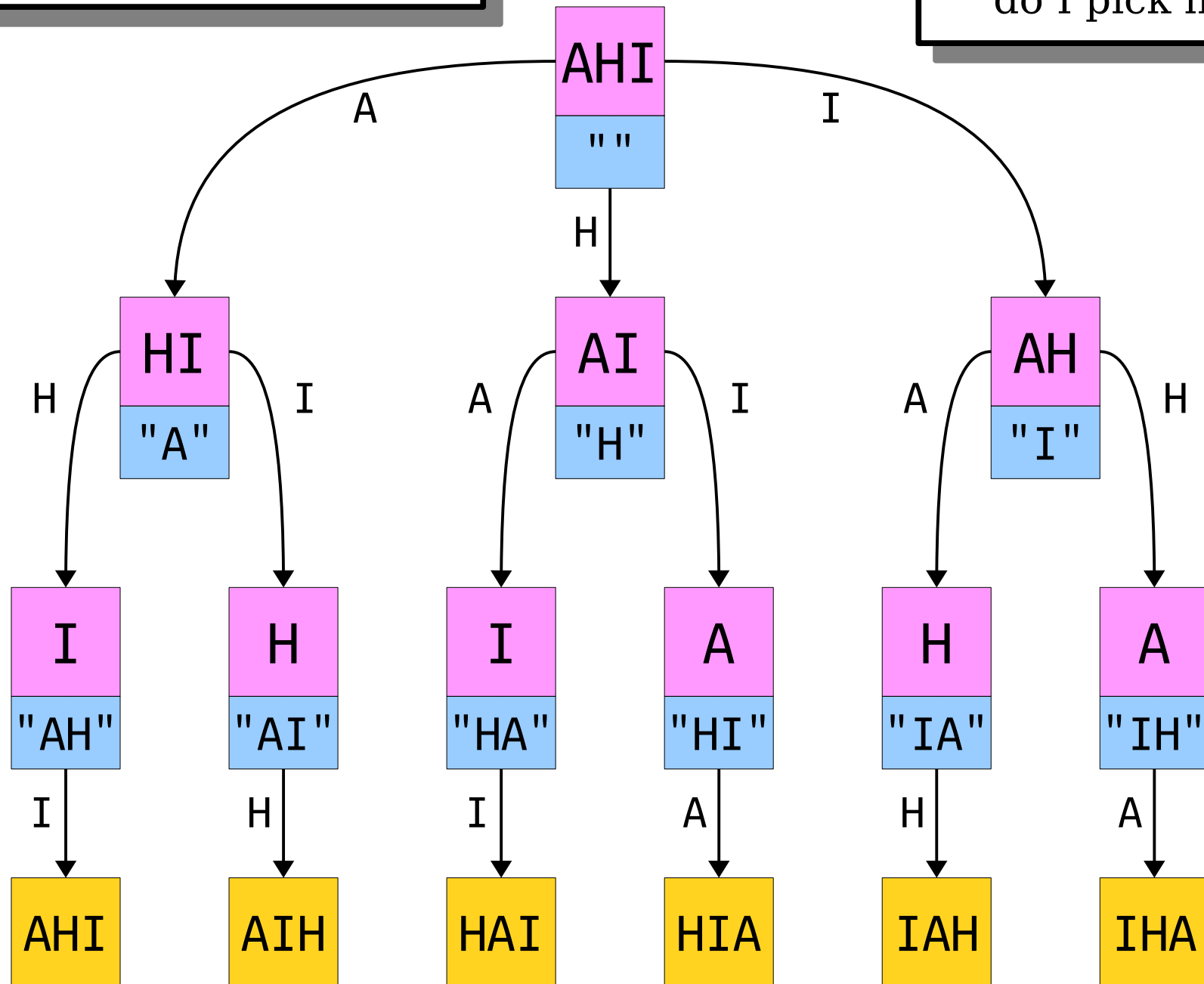
Lava Beds National Monument

List all *permutations* of
 $\{A, H, I\}$

Each decision is of
the form “which item
do I pick next?”

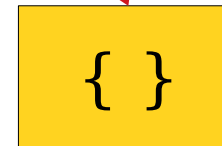
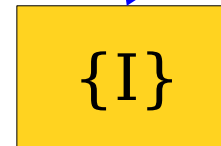
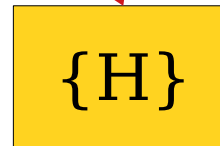
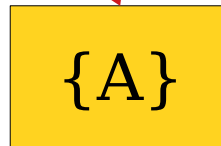
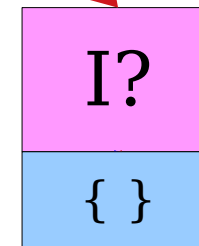
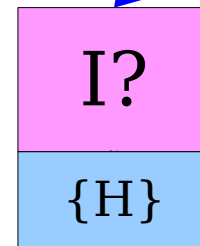
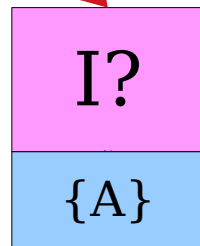
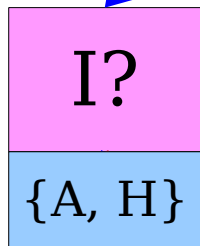
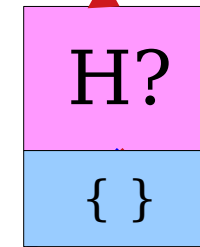
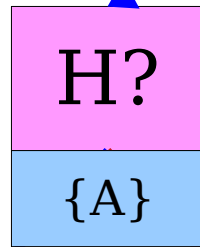
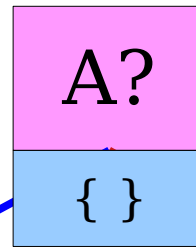
List all *permutations* of
{A, H, I}

Each decision is of
the form "which item
do I pick next?"



List all *subsets* of
 $\{A, H, I\}$

Each decision is of
the form "do I
include this item?"



Storing Permutations

```
Set<string> permutationsOf(const string& str);
```

Summary for Today

- Recursion and iteration aren't mutually exclusive and are frequently combined.
- We can enumerate subsets using a decision tree of “do I pick this?” We can enumerate permutations using a decision tree of “what do I pick next?”
- Recursive functions can both print all objects of some type and return all objects of some type.

Your Action Items

- ***Read Chapter 8***
 - There are so many goodies there, and it's a great way to complement what we're discussing here.
- ***Work on Assignment 3***
 - Aim to complete the Sierpinski Triangle and Human Pyramids by our Monday lecture.
 - If you have time, start tinkering around with "What Are YOU Doing?"

Next Time

- ***Enumerating Combinations***
 - Can you build the Dream Team?
- ***Recursive Backtracking***
 - Finding a needle in a haystack.
- ***The Great Shrinkable Word Problem***
 - A fun language exercise with a cute backstory.