

## SYNOPSIS

- Your first recursion lab!

## GETTING STARTED

- Run the following commands:

```
mkdir obj bin src
```

```
cp -r ~/jplank/cs140/Labs/Lab9/txt .
```

```
cp ~/jplank/cs140/Labs/Lab9/makefile .
```

```
cp ~/jplank/cs140/Labs/Lab9/bin/ss_random_game bin
```

```
cp ~/jplank/cs140/Labs/Lab9/bin/ss_player bin
```

- I have created a [hints page](#) strictly for ShapeShifter.  
You may access it here:

- <http://uth.claranguyet.me/lab.php?id=cs140/sp20/lab9>

It features some pretty nice pseudocode with **A LOT** of comments to push you to success.

- **START EARLY.** Solving these iteratively is trivial. But you must use recursion. Debugging recursion can be tricky.

- Pseudocode for [enum](#) on next page. Enjoy!

## SUBMISSION COMMAND

- `tar -cvf lab9.tar src/enum.cpp src/ss_solver.cpp`

## PART I - Enum

- A few ways to do this:

### John's pseudocode:

# Base Case

if index == s.size():

if ones == 0:

# Valid bitstring

print(s)

return

else:

# Invalid bitstring

return

# We're not at the end of the string yet

# Keep trying things

if ones <= remaining characters:

s[index] = '0'

do\_enum(index + 1, ones)

if ones > 0:

s[index] = '1'

do\_enum(index + 1, ones - 1)

else

# There were more ones than elements remaining

# Exit early

return

### Clara's Pseudocode

# Hit the end of the string

if index == s.size

print(s);

return

# Initially set to '0'

set s[index] to '0'

# If there are ones to spare and we

# are toward the end, set to 1 instead

if (there are ones left and s.size() - index ≤ n\_ones):

set s[index] to '1'

# Unleash recursion

if (s[index] is 1):

do\_enum(index + 1, num - 1) # a '1' was added

else:

do\_enum(index + 1, num) # a '0' was added

# Well... if it isn't already a '1' and we have spares, set

if (s[index] is not '1' and n\_ones > 0):

set s[index] to '1'

do\_enum(index + 1, num - 1)

# Ok... go back to -

set s[index] to '-'

return

## PART II - SHAPESHIFTER

- Honestly, look at guide. It's better than reading my handwriting... 😅