Class 17

Recursion

Recursion is a programming technique

Some Pros:

1. The code may be easier to write.
2. Some problems are inherently recursive.
3. Recursion reduces the length of the code.

Some Cons:

1. Recursive functions are generally slower than non-recursive ones.
2. They may require a lot of memory to hold intermediate results.
3. Recursive code can be hard to debug.

Example 1

- fibonacci
  - compute the n^{th} Fibonacci number

Process of writing a recursive function

1. Break the problem to solve down into a problem that is one step simpler
1. Assume the function will work to solve the simpler problem
2. Since we know we can solve the simpler problem, how would we use this to solve the more complex problem?

Example 2

- update even digits of an int
  - add 1 to all even digits of a number and return modified integer

Example 3

- countdown
  - recursively print integers n through 1

Example 4

- has3
  - determine whether an integer contains a 3 in it

Example 5

- print_triangle
  - recursively print a triangle

Example 6

- recursiveSum
  - recursively sum integers 1 through n

Example 7

- recursiveMult
  - compute the product of two positive integers using recursion

Example 8

- findPower
  - compute an integer raised to a power using recursion

Example 9

- sumDigits
  - recursively sum the digits of an integer