CSE 114A - lecture 6!

last time -

built-in datatypes
like lists, tuples, Ints, Strings, Bools...

today, and next time -

- user-defined data types
- more discussion of recursion, especially <u>tail</u> recursion.

Haskell gives you several ways to define your own data types:

- Product types -
  for putting together values of several types:
  a value of type T can contain values of types T_1 <u>and</u> T_2.
  (Q: why not just use tuples?)
- Sum types -
  for types that are one of multiple options:
  a value of type T can be a value of T_1 <u>or</u> T_2.
- Recursive types -
  for inductively defined data
  a value of type T can contain a sub-value that's itself of type T.

These mechanisms can all be combined and used together to express rich, interesting types.

↑ example: we want to represent programs, so we can operate on them!

We can represent programs as <u>abstract syntax trees.</u>

↑ for this we'll need all of the above mechanisms!