Lists

- Have already seen them before: A = [1, 2, 3, 4]
- O(1) access, like arrays, O(1) appends, O(1) to remove from the end
- O(n) insert/delete if not at the end (if deleting from the front, need to shift n elements backwards to fill those indices)
- Implemented using fixed-size arrays, and backing arrays are reallocated when necessary
    - Similar idea to sets

Tuples

- Example (multiple elements): T = (1, 2, 3)
- Example (single element): T = (1,)
- Like lists, but **immutable** - cannot be modified once they are created
- Tuples are immutable, but elements can be pointers to mutable data
    - Example: T = ([1, 2], 3)
    - Therefore, not guaranteed to be hashable