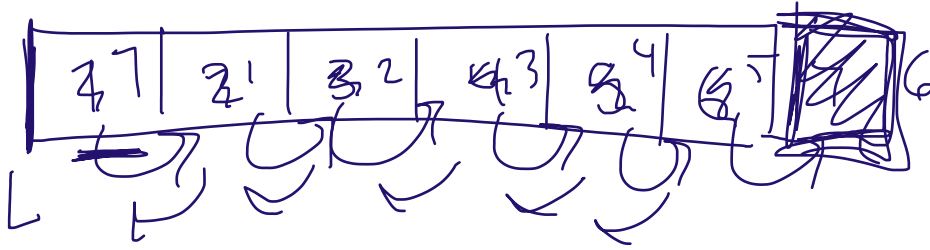


Python list



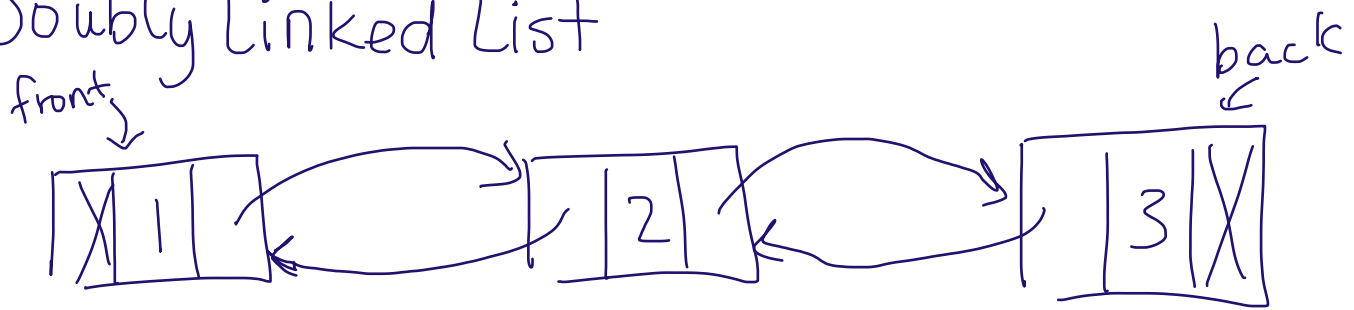
`L.append(7)`
`L.pop()` } very fast

`L.insert(0, 7)`
`L.pop(0)` } if we have a big list → slow

LLs:

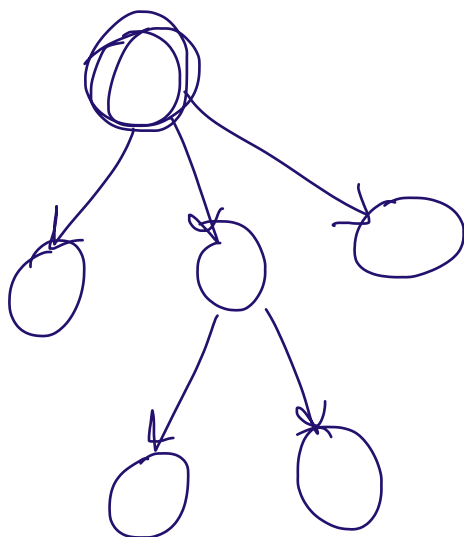
prepend →
append →
delete-front → } fast, just look at self.front/self.back
delete-back → slow, bc need to find 2nd last

Doubly Linked List



- extra overhead
have to maintain prev ref as well
- but → could make delete-back fast too.

Linear Data Structures



Trees!