Memory Management

Soundness — don't free if needed

Completeness — free when not needed

↳ time — how cpu is taken by M, M, from program

↳ space — how far from optimal the memuse is ↳ % ↳ pauses

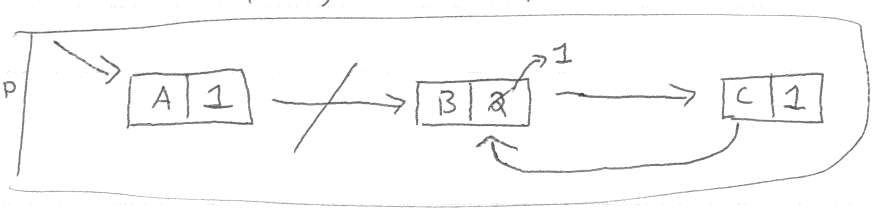
↳ absolute

Manual (ie malloc/free) — not sound (mistakes)

— tends to have 2x space + lg n malloc/free

Smart Pointer := ptr + counter — Reference Counting

- malloc initialize the count to 1
- every time you make an alias (ie a copy of the pointer) increment count
  ↳ retain
- when you destroy an alias, decrement → release
- if after dec, count is 0, then free
- trivial programs retain/release can be sounded inserted
  ↳ therefore, programmers must insert for some programs ⇒ ¬ sound



- cyclic data are incomplete (ie leak memory)
- space := count takes space (25% — 50% depending on structs) (if word)
  if byte, 3% - 6% → limits the count → leaks (overflow)
- time := m/f are still lgn + bookkeeping (α to your work)
- cascading deletes → large pauses