

# Quiz 2.1: hand-over-hand

- Check if **contains** requires locking
  - ✓ What if **contains** traversed the list without lock acquisition?
- What if traverse (in remove, insert) checks the value in **curr** before locking it (only holds lock on **pred** when traverse terminates)?
- Can we just use one lock at a time?
- Prove starvation-freedom (assuming starvation-free locks)
  - ✓ Can an operation be blocked (delayed forever) by infinitely many concurrent inserts?

# Quiz 2.2: optimistic

- Show that validation is **necessary** for updates
  - ✓ Hint: consider an algorithm without validation and show that an update can get **lost** because of a series of concurrent removes
- Is validation necessary for contains?
- Show that the algorithm is **not** starvation-free (even if all locks are)

# Quiz 2.3: lazy

- Show that both conditions in the validation check are **necessary**

Hint: consider concurrent **removes** on two consecutive nodes, or a **remove** concurrent to an **insert** of a preceding node

- Is the check **!curr.marked** necessary in **contains**?

- Determine linearization points for all operations:

- ✓ **insert**(successful or not)
- ✓ **remove** (successful or not)
- ✓ **contains** (successful or not)

Hint: for an unsuccessful **contains**(x), linearization point may vary depending on the presence of a concurrent **insert**(x)