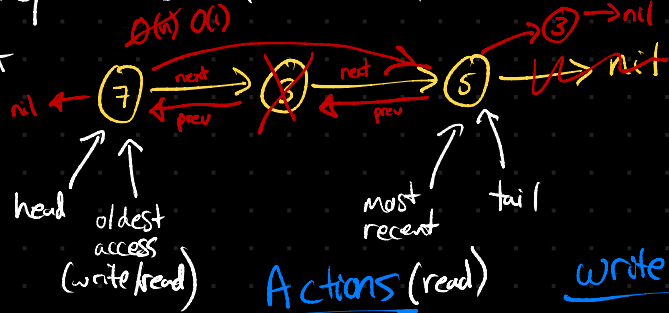


# Least Recently Used (LRU) Cache

Link to starter code below



- timestamps → slow  $O(n)$  search
- dictionary → cache  $O(1)$  look + insert
- linked list



queue (last in  
last out)  
- drop last

## Flow

1. Cache (Dict)
2. LRU enhancement
  - edge cases
  - DLL

## Actions (read)

1. delete
2. insert

- toString()
- tests to validate

## write actions

1. insert (append)