LRU CACHE - LEAST RECENTLY USED.

t₁ t₂ ... t_t t_t remove least recently used Time Stamp Full gremove t, Helpers: · Limited Memory Retrieval O(1) · Remove LRU get(K) put (Kv) add Note (Note) @ head How about putting elements always to front? groman Node (Node) When get hugged that element should be most recently used move to Kead (Node) gemane LRU() Hash Table - O(1) Key exist flow to maintain ordering -LRU? case: !exist move To Head () node = new () gretum Null? Linked Hash Map map. put (K, V) case: exists why not singly 99 deletien @ add Node (node) > Double Linked List: Order node ←map.get (key) more Totted (Hade) Case: capacity maxed Hash Map: Efficient LOOK up seturn rade removeLRU Head Doubly Linked List Tail Head ZK, V ZK, V Zail Most HashMap Recently Used.

Capacity Size

Yash Pradhan Study Notes