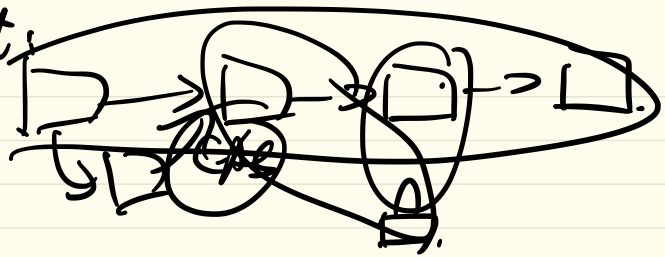
Set:



Linked List

I.

- add()
- remove()
- contains() → wait-free.

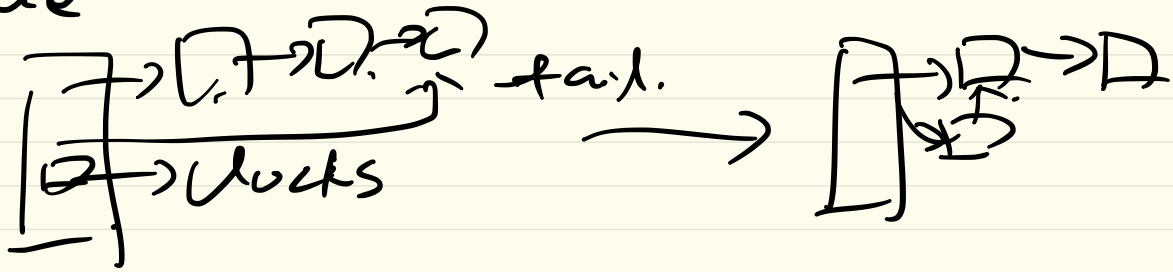
→ fine-grained

II

- add()
- remove()
- find()
- contains() → wait-free.

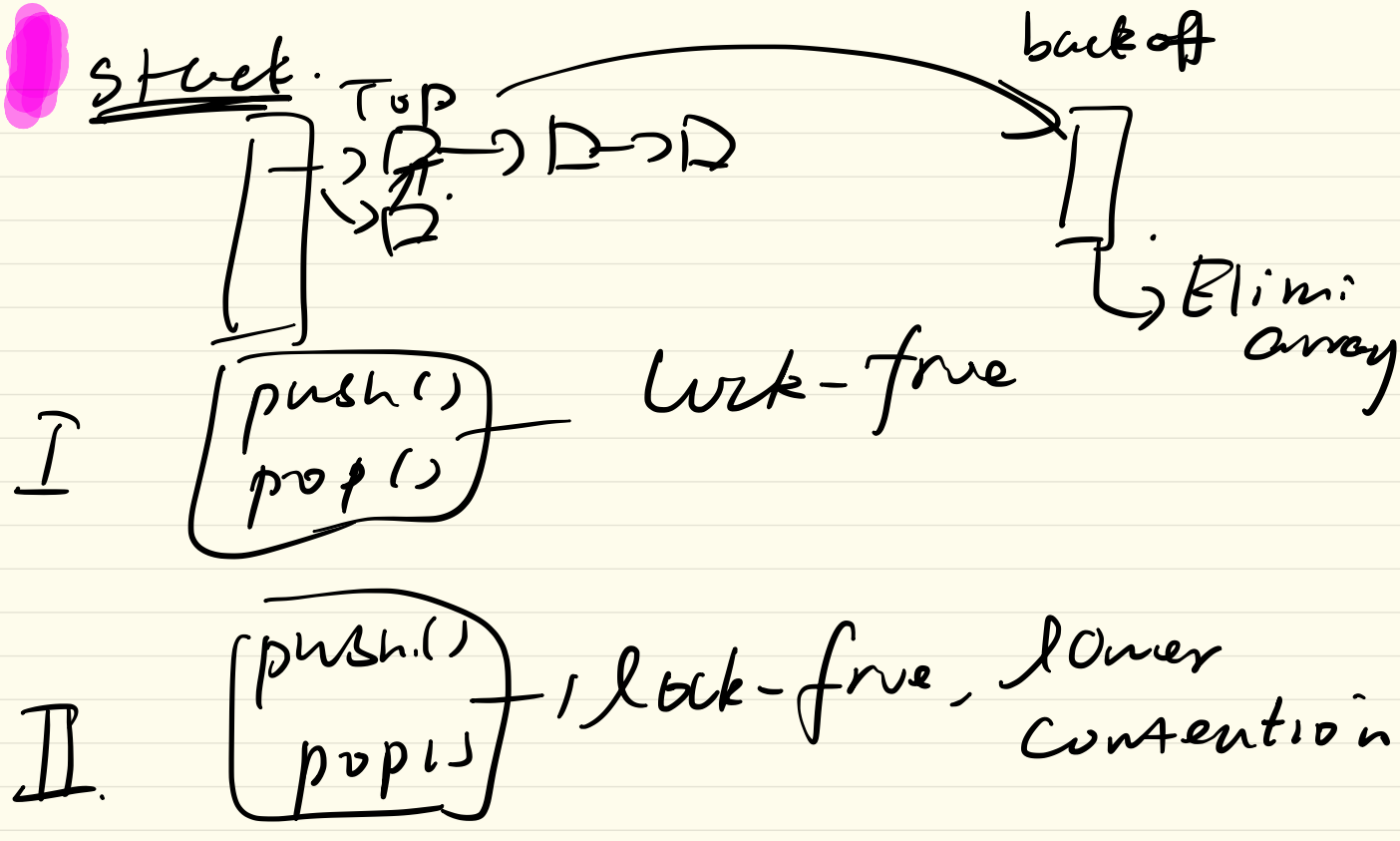
→ lock-free

queue

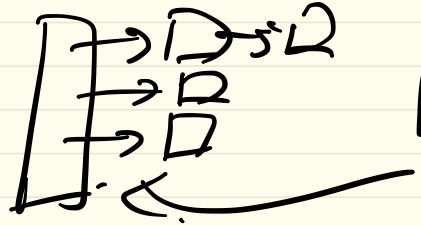


I. : enq() deq() → fine-grained lock.

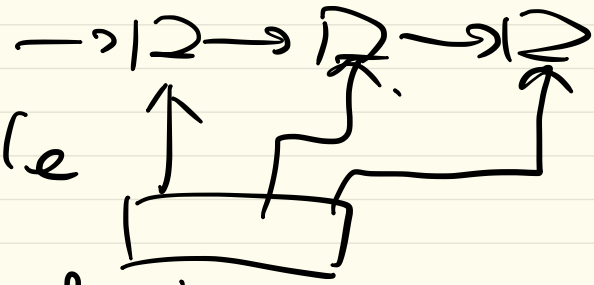
II. : enq() deq() → lock-free.
(not wait-free?)


Stack
Top
back off
Elimi
array
I
push()
pop()
Lock-free
II
push()
pop()
lock-free, lower
contention

hashing



hashtable



ktable.

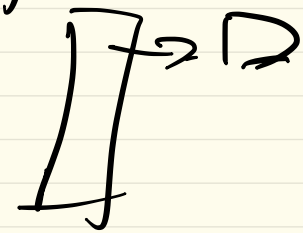
I
- add()
- remove()
- contains()
- ~~resize()~~

→ fine-grained

→ wait-free (Lazy)

II
- add() → lock-free
- remove() → lock-free
- contains() → wait-free
- resize() → lock-free!

Snapshot

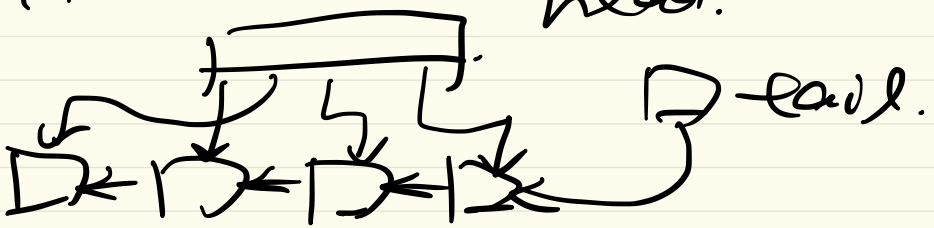


MRSW.

I. [update() → wait-free
scan() → lock-free]

II. [update() → wait-free
scan() → wait-free.]

Universal.

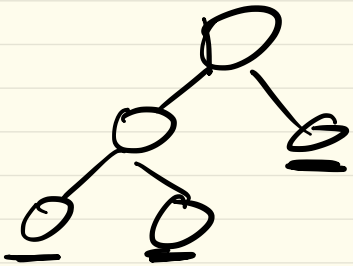


I apply() → lock-free

II apply() → wait-free

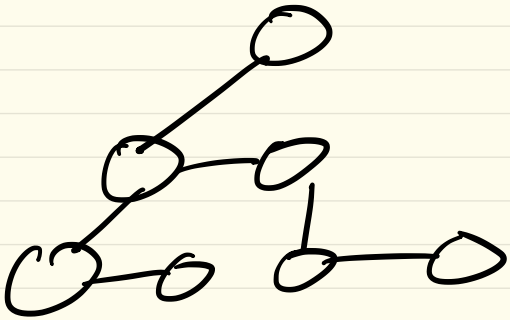
barrier

I. CBT:



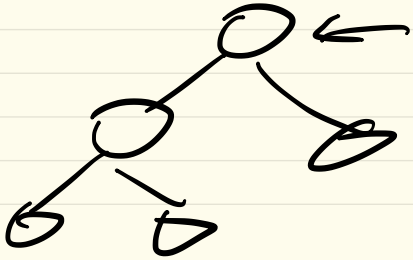
fewer contention

II. Tournament Tree:



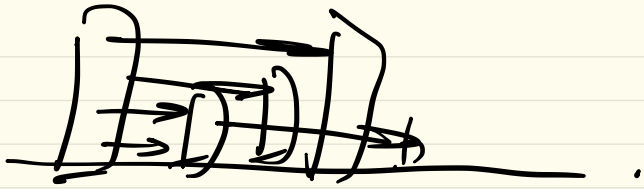
fewer

III. Static Tree



thread.

Skip List



I. add()
remove()
contains()

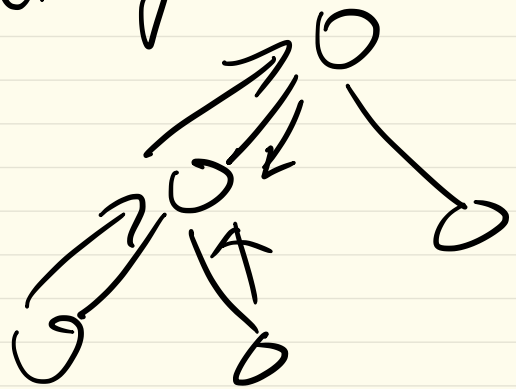
→ optimistic Lazy lock

→ wait free

X.

Counting

CBT



fine-grained locks.

getAndIncreamount()

x 2