1. Smallest Substring with all characters
2. LRU Cache
3. Longest Subarray with equal freq of 0, 1 and 2
4. Snapshot Array

Agenda

Longest Subarray with equal freq. of 0's, 1's and 2's

int[] arr = { 1, 1, 0, 2, 1, 0, 1, 2, 1, 2, 2, 0, 1 }

0's → 1
1's → 1
2's → 1

len = 3

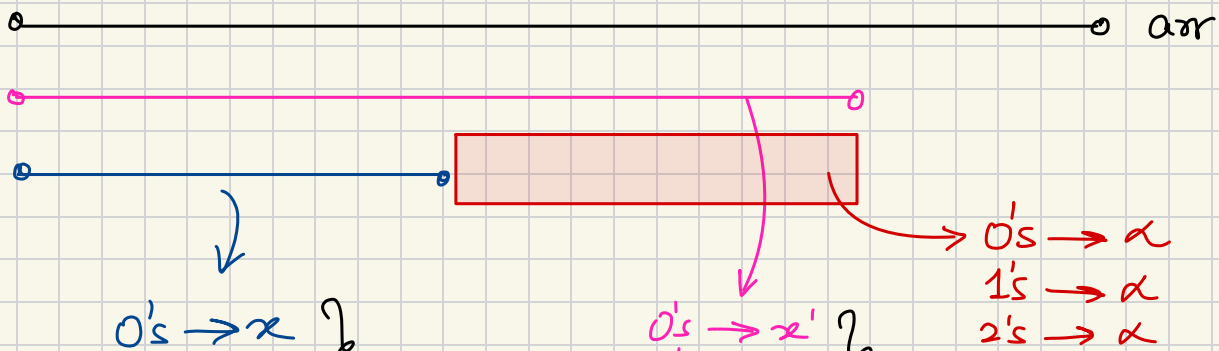
Brute Force

Find all the subarrays, and partially count- freq 0's, 1's and 2's

- store the longest len.

TC: O(N^2) SC: O(1)

int[] arr = { }



$x' = x + \alpha$

$y' = y + \alpha$

$z' = z + \alpha$

$\Rightarrow y' - x' = y - x$

$z' - y' = z - y$

int[] arr = { 1, 1, 0, 2, 1, 0, 1, 2, 1, 2, 2, 0, 1 }

len = 6

0's x	0	0	0	1	1	1	2	2	2	2	2	2	3	3
1's y	0	1	2	2	2	3	3	4	4	5	5	5	5	6
2's z	0	0	0	0	1	1	1	1	2	2	3	4	4	4
y-x	0	1	2	1	1	2	1	2	2	3	3	3	2	3
z-y	0	-1	-2	-2	-1	-2	-2	-3	-2	-3	-2	-1	-1	-2

0#0 1#-1

2#-2

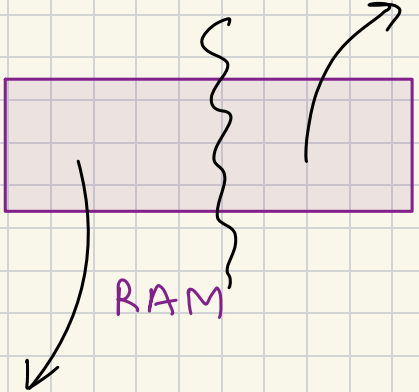
HashMap

key index

2 Keys X

encoding = y-x # z-y {String}

LRU Cache

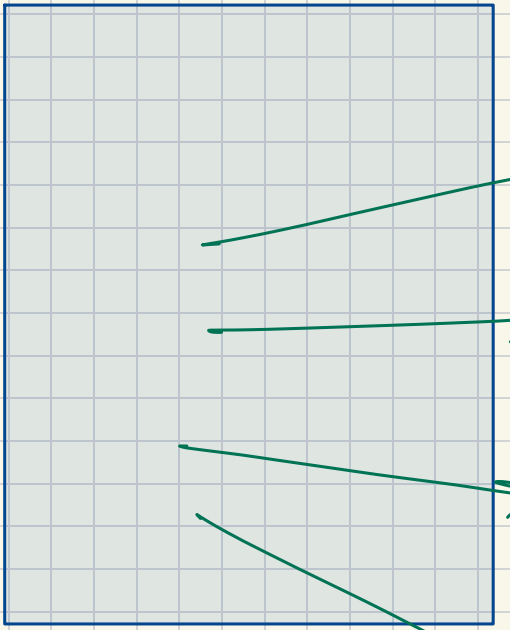


Cache Memory

{ Apps running in background }

Main Memory

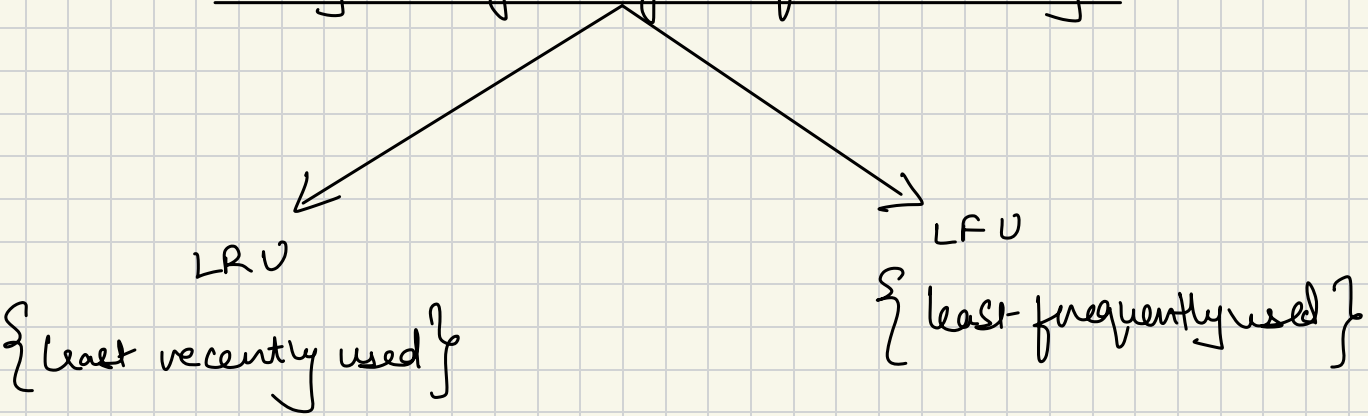
{ apps visible onscreen }



App1, App2, App3, App4

Cache { 3 App }

Memory Managment System for Cache Memory



0's App1 → opened

2's App2 → opened

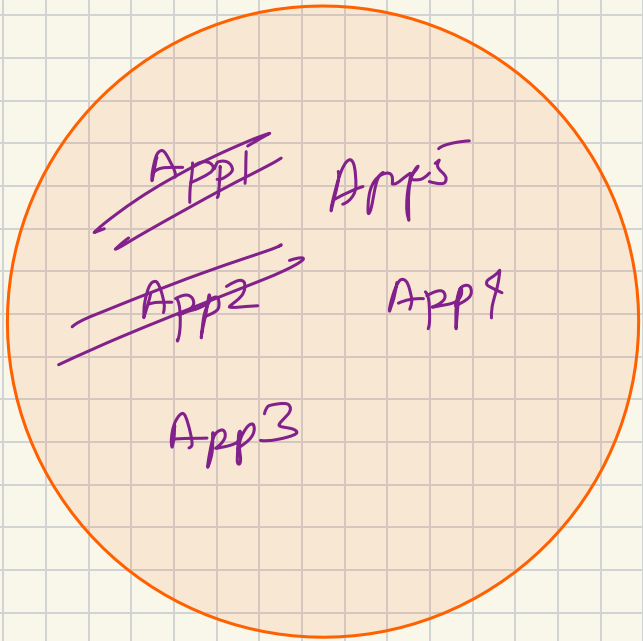
5's App3 → opened

6's App1 → notification

7's App4 → opened

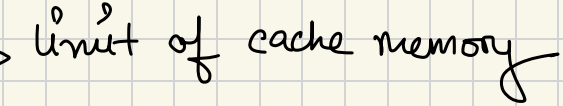
8's App3 → reopened

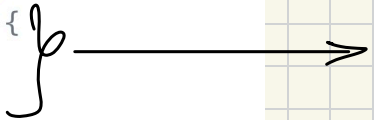
9's App5 → opened

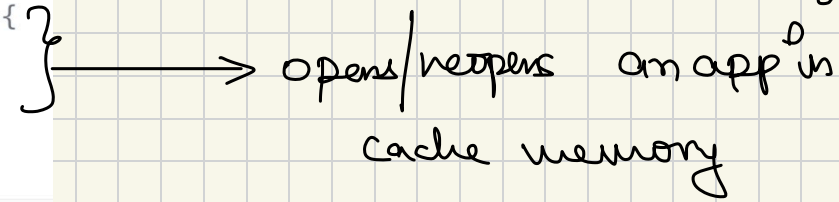


Cache Memory
{3 applications}

```
class LRUCache {
    // your code here
    public LRUCache(int capacity) {
        // your code here
    }

    public int get(int key) {
        // your code here
    }

    public void set(int key, int value) {
        // your code here
    }

}
```







LRUCache(3)

set(1, 10)

set(2, 30)

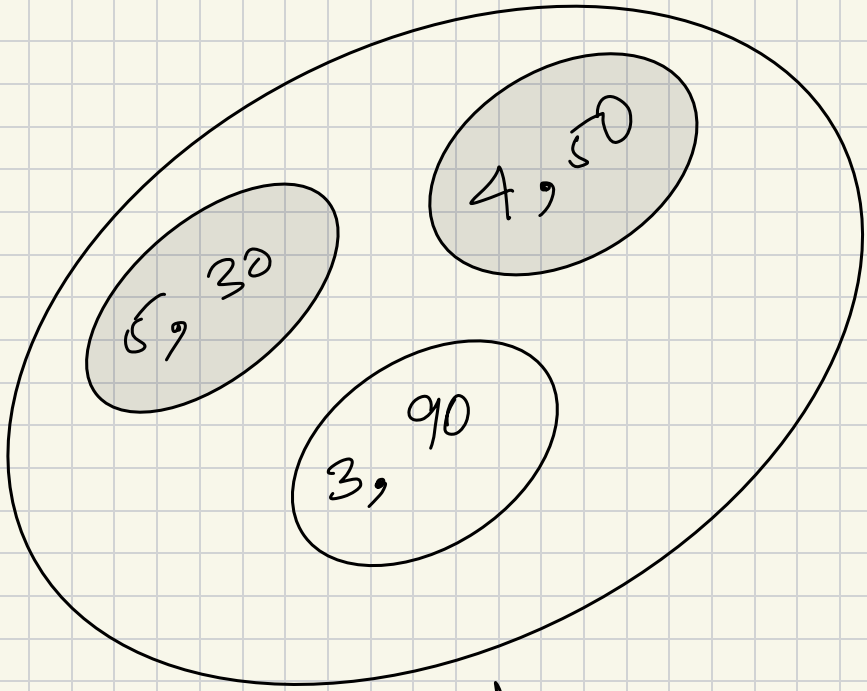
set(3, 40)

get(1) ⟶ 10

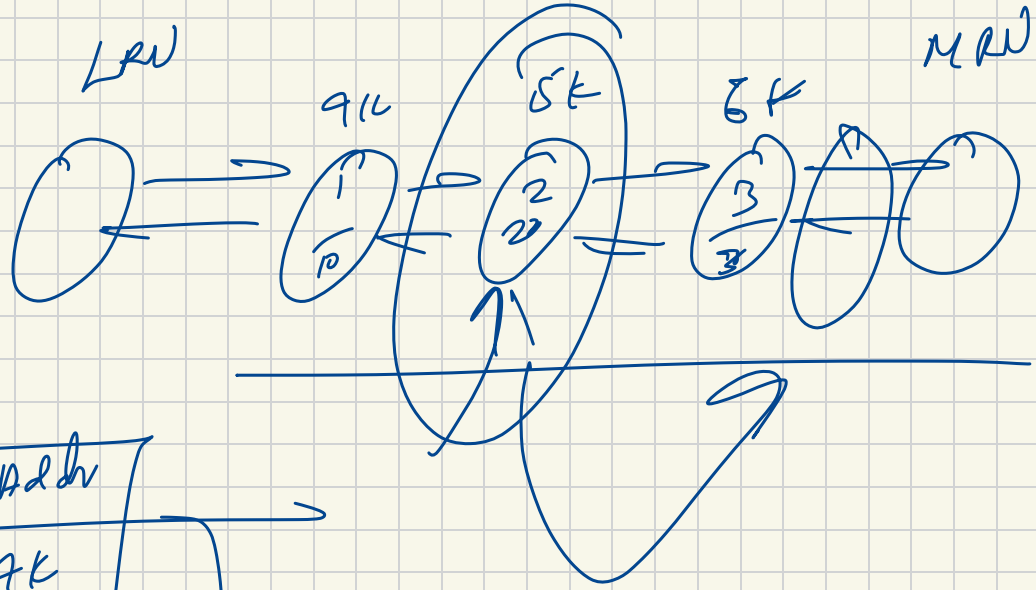
set(4, 50)

set(3, 90)

set(6, 30)



doubly linked list



Data	Addr
1	9K
2	5K
3	6K

Snapshot Array

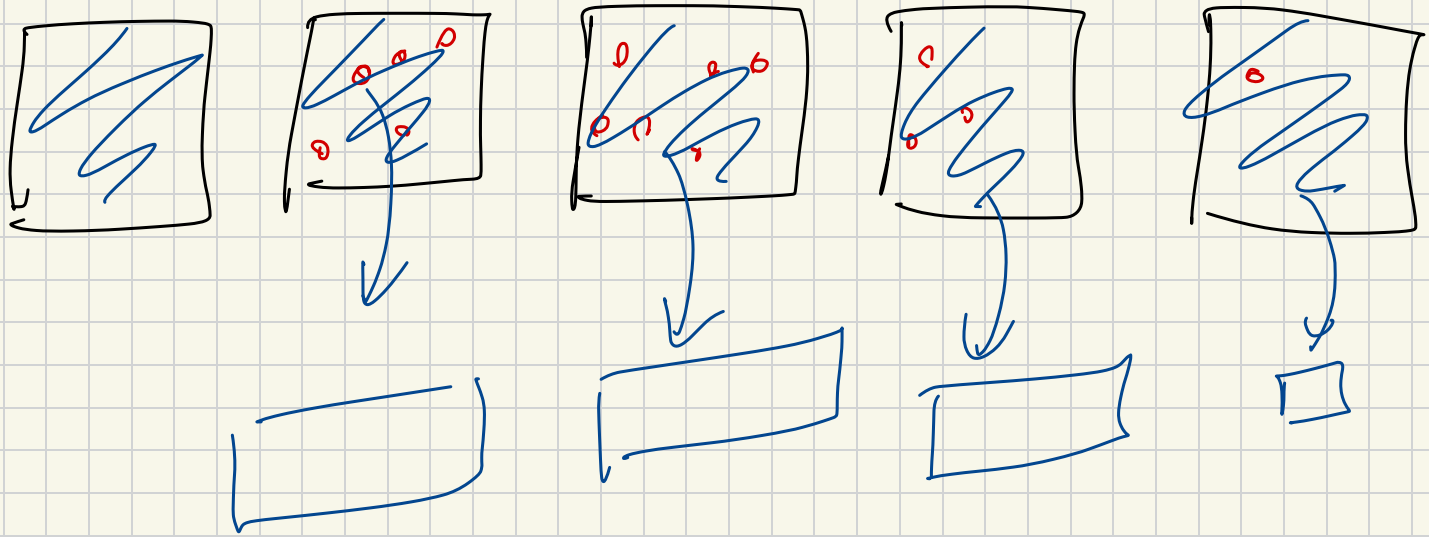
int[] = { 0, 9, 20, 0, 0, 0, 7 }
(0 1 2 3 4 5 6)

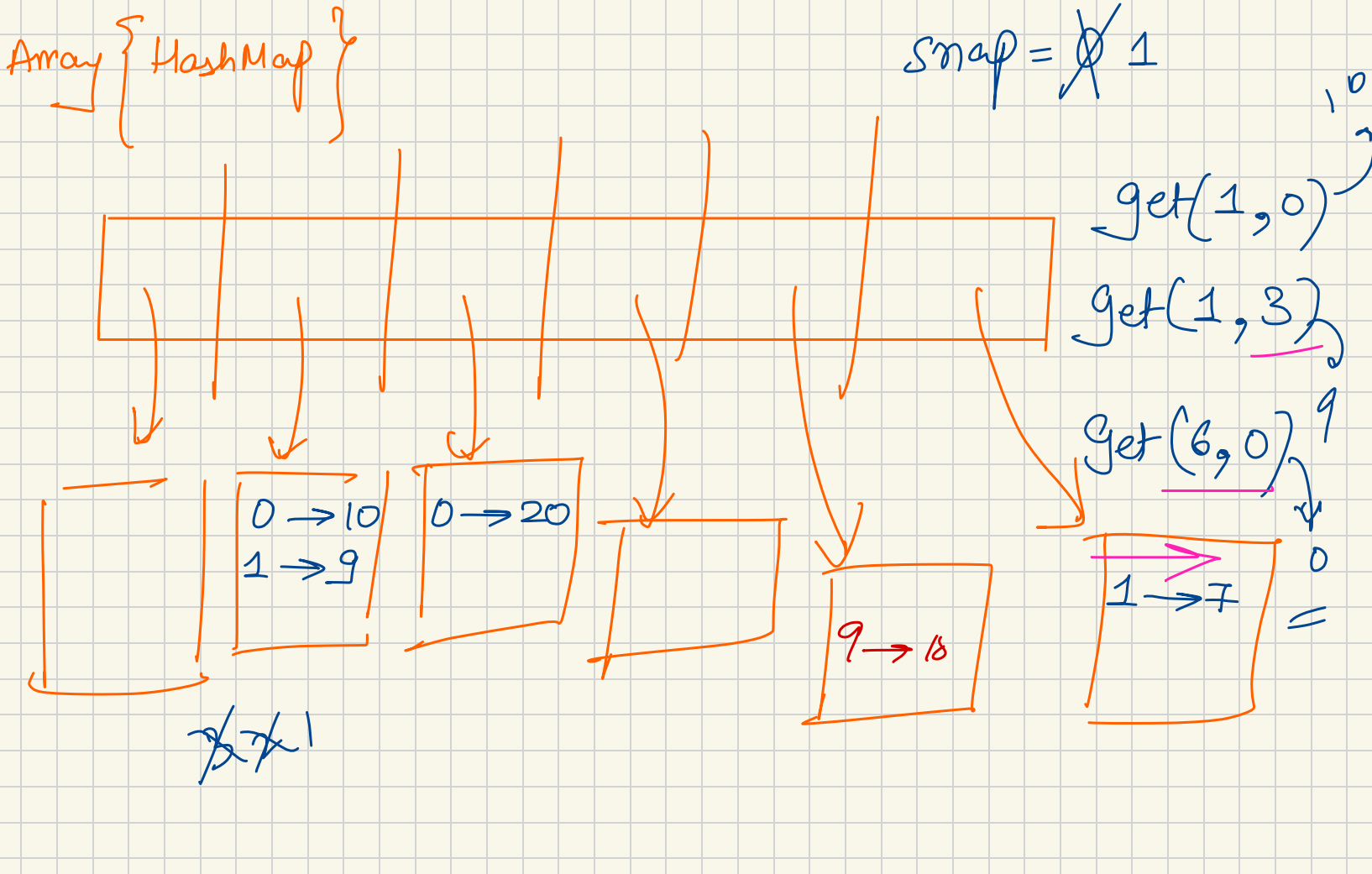
set(1, 10)
set(2, 20)
snap()
set(1, 9)
set(6, 7)
snap()
get(1, 0) ⇝ 10

Key	AL
snap = 0	{ 0, 10, 20, 0, 0, 0, 0 }
snap = 1	{ 0, 9, 20, 0, 0, 0, 7 }

get(2, 5) ⇝ 20

get(6, 0) ⇝ 0
get(6, 2) ⇝ 7




Array { HashMap }
snap = 1
get(1,0)
get(1,3)
get(6,0)
0 → 10
1 → 9
0 → 20
9 → 18
1 → 7

int[]

String[]

Pair[]

HashMap[]