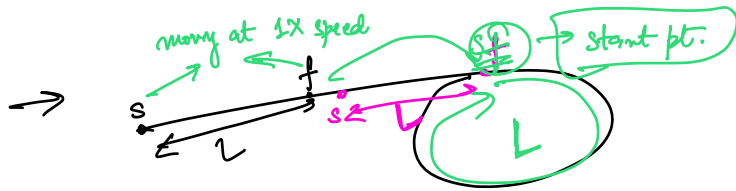


→ figuring out if loop exists.
 → meeting point of S & f.

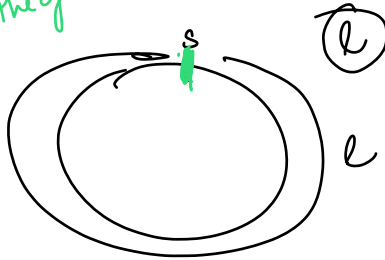




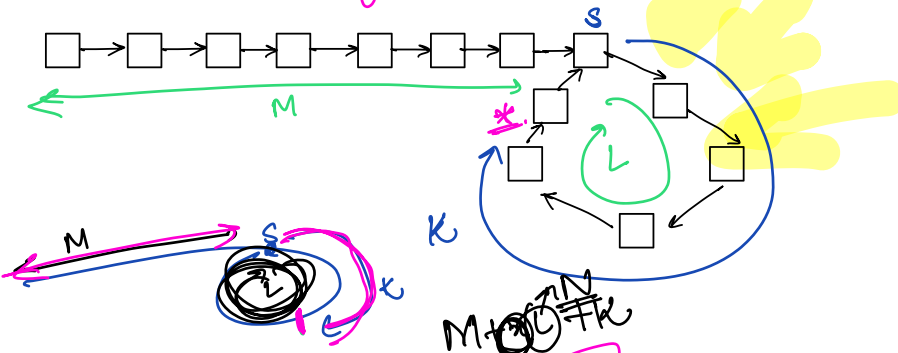
Approach 2
to find start pt. in the cycle

If start at s:
 $l : s$
 $2l : s$
 $3l : s$

$Nl : s$



Distances covered by each ptr. before meeting.



$$\begin{aligned} d(\text{fast}) &= M + xL + K \\ d(\text{slow}) &= M + yL + K \end{aligned}$$

$$S = \frac{d}{L}$$

$$\text{speed}(\text{fast}) = 2 \times \text{speed}(\text{slow})$$

$$d(\text{fast}) = 2 \times d(\text{slow})$$

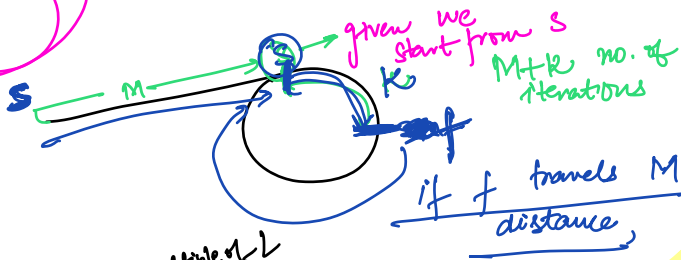
$$M + xL + K = 2M + 2yL + 2K$$

$$(x - 2y)L = M + K$$

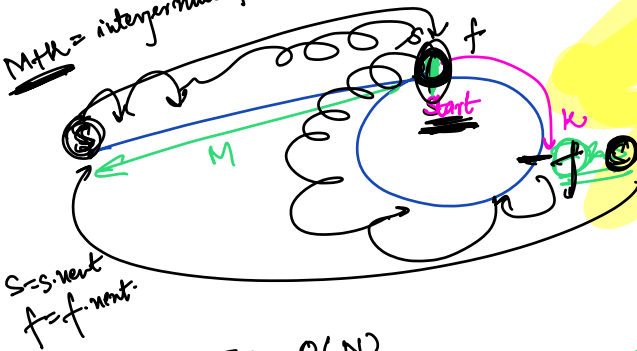
$$L = \frac{M + K}{x - 2y}$$



($M+K$ is an integer multiple of L)

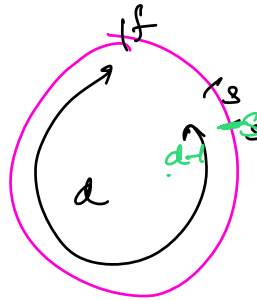
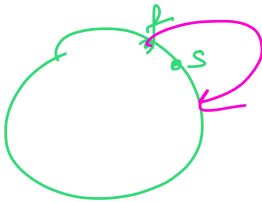


$M+K = \text{integer multiple of } L$



T.C = $O(N)$

$d = M + K + \alpha L$

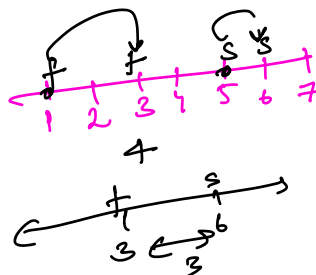


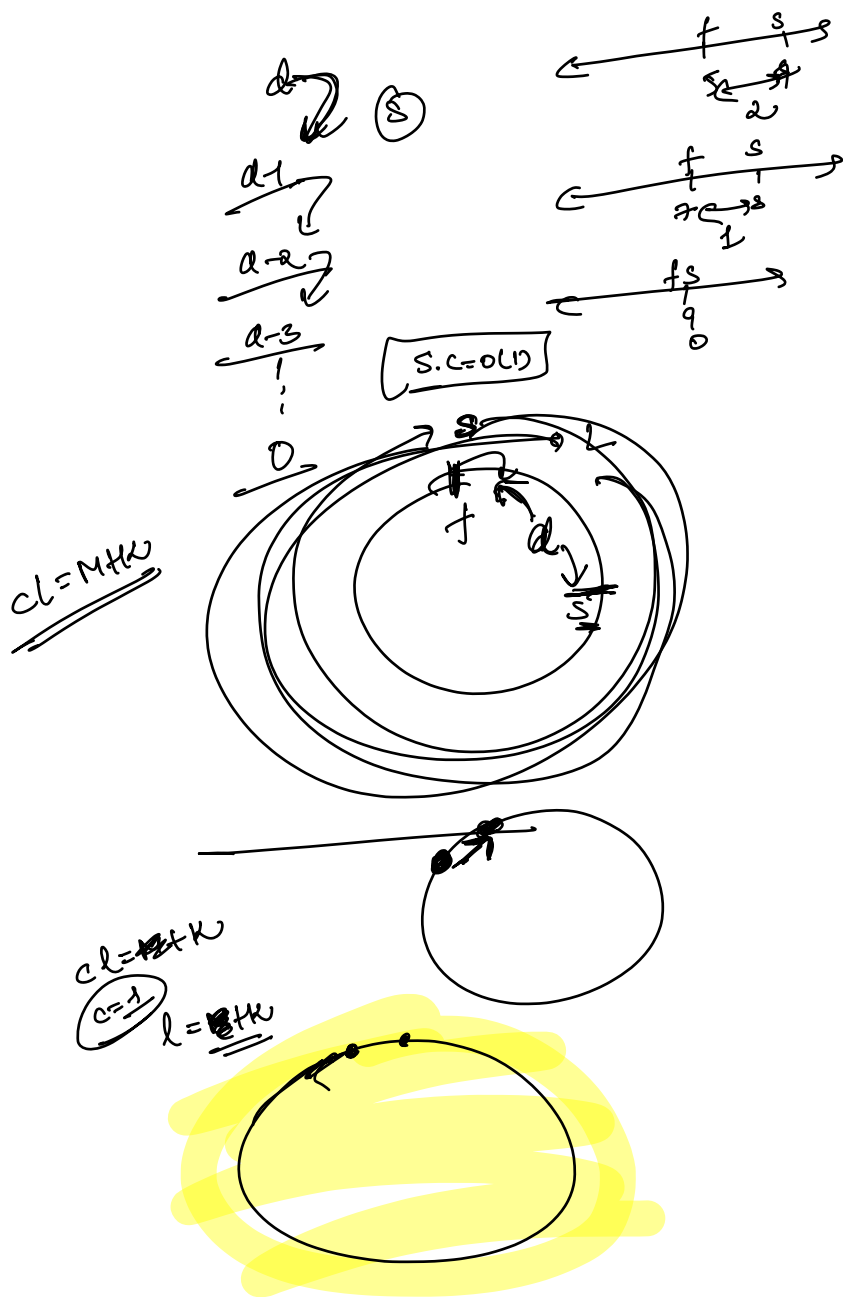
(d)

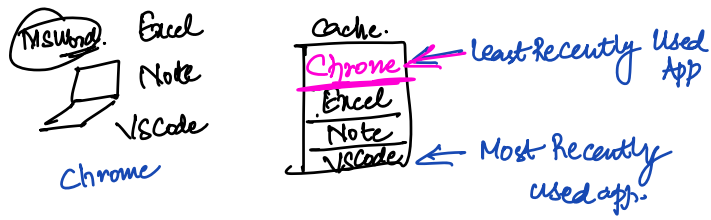
slow : $\underline{\underline{d+1}}$
fast : $\underline{\underline{d-1}}$

$f \xrightarrow{d+1} s$
 $(d+1) - 2 = \underline{\underline{d-1}}$

$d-1$
 $d-2$
 $d-3$

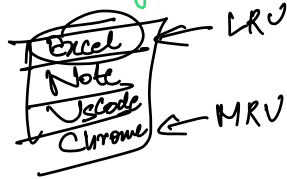




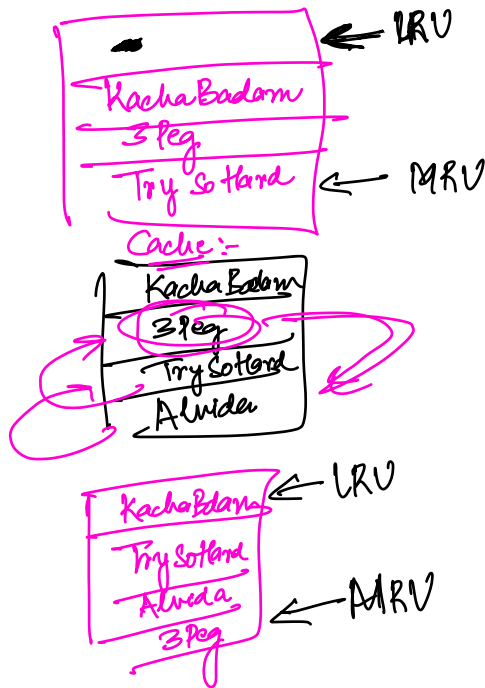


LRU
↓ ↓ ↓

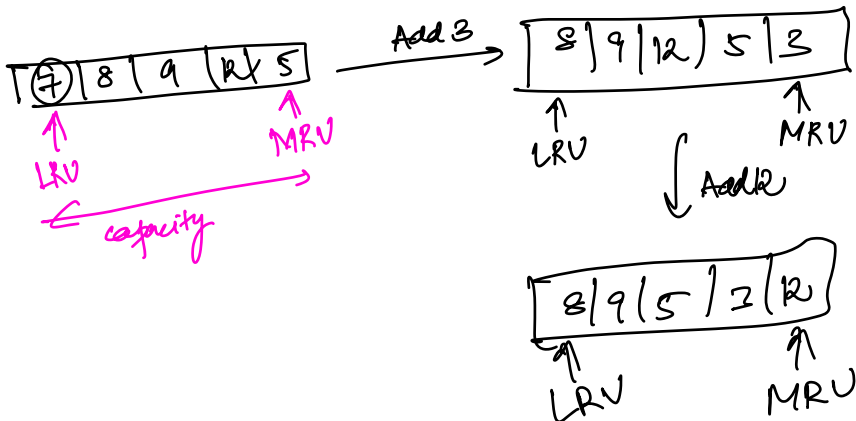
Eviction Policy

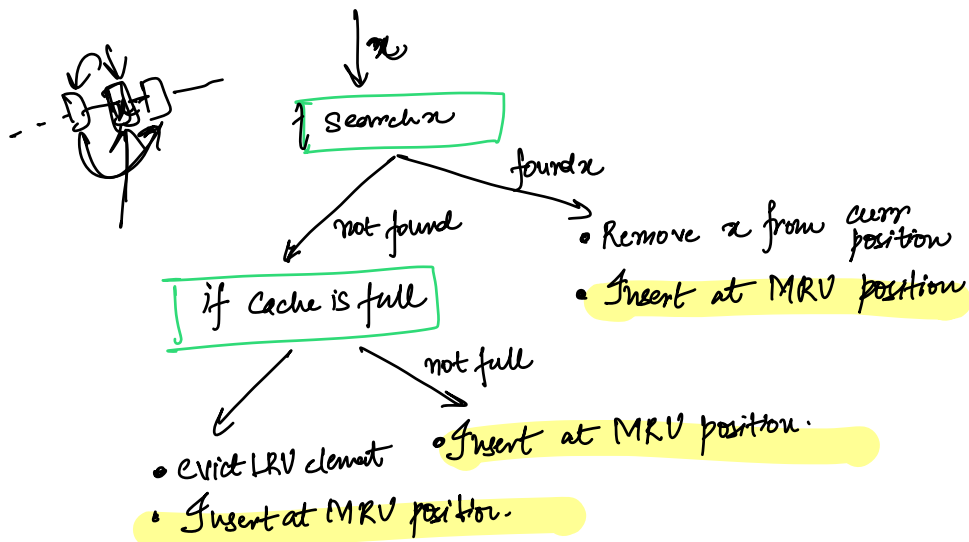


Spotify
Alvida.



7, 8, 9, 12, 5, 3, 12.





Search
delete
insert

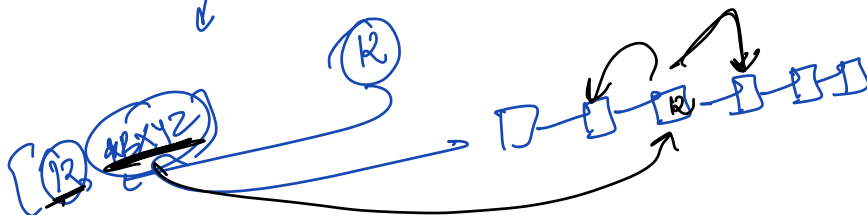
- insert
- search
- delete

arrays	Linked list
$O(N)$	$O(N)$
$O(N)$	$O(1)$
$O(1)$	$O(1)$

Note: In the linked list column, the $O(N)$ is circled in pink, and a pink arrow points to the $O(1)$ below it with the text "done after search".

To optimize search \rightarrow Hashmap

Hashmap \langle data, add \rangle



Doubly
L.L

class DLL {
int data;
DLL next;
DLL prev;

