CSE321: Operating Systems

Topic: Virtual Memory

Prepared by:
Saad Bin Sohan

BRAC University

Email: sohan.academics@gmail.com
GitHub: https://github.com/saad-bin-sohan

Virtual Memory

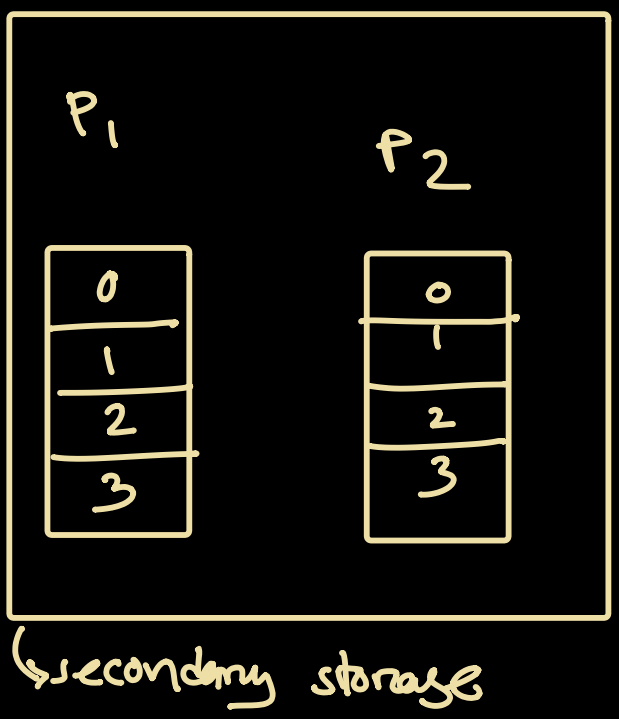
→ when the program size is larger than the main memory size or remaining free space on it, memory spaces are allocated on the secondary storages and it gives the program an illusion that space on the main memory has been allocated. That's called virtual memory.

- valid / invalid bit : frame এর corresponding existance যদি RAM এ থাকে, then the bit is 1. না থাকলে 0.

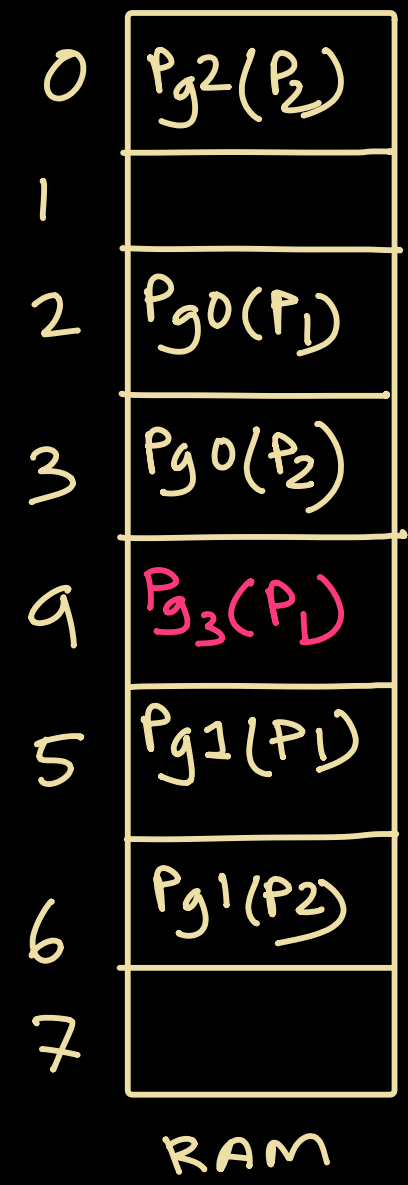
i.e:

Page table (P_1)

P#	F#	valid/invalid bit
0	2	1
1	5	1
2		0
3	4	~~0~~ 1



main memory



free frames

1

~~4~~

7

Demand Paging: when CPU looks for the F# using the P#, that lookup is called Demand Paging.

2 consequences of demand paging:

i) Page hit: during lookup, if the valid/invalid bit is 1 meaning it's frame exists on the RAM, it's a Page Hit.

ii) Page Fault: if the bit is 0 meaning absent in RAM, it's called a Page Fault.

what happens if it's a Page Fault?

ans:

step 1) the lookup for bit is done at user mode

step 2) if invalid, a trap/interrupt is generated by the OS (Kernel mode)

3) OS ই particular process এর জন্য secondary storage এ logical address এ lookup করে।

4) free frame খুঁজে এবং ই Page কে map করে।

→ this is called 'swap in'

5) and the page table will get updated.

Important terminologies:

i) valid / invalid bit

ii) Demand paging

iii) page hit

iv) page fault

v) swap in

vi) swap out: RAM ২ existing page কে remove করে দেওয়া (when no frame

is free)

page replacement = swap out + swap in

Algorithms of page replacement:

i) FIFO (First In First Out) Page Replacement Algorithm:

→ যে page সবার first এ RAM এ entry নিয়েছিল, সে সবার আগে leave নিবে।

Que এ given থাকবে –

i) number of frame (i.e: 3)

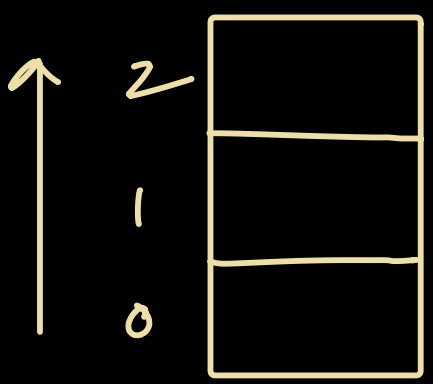
11) Reference string:

(i.e: 7, 0, 1, 2, 0, 3, 0, 4, 2, 3, 0, 3, 1, 2, 0)

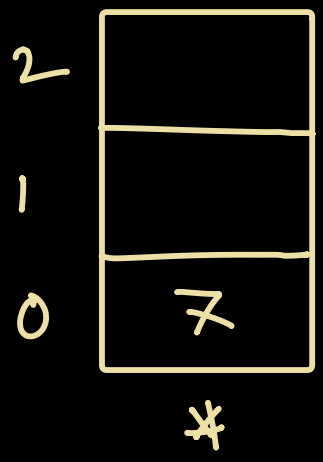
CPU এই moment এই page টাকে execute করতে demand করছে

বা A, B, C, D ... এগুলো চাকতে থাকে।

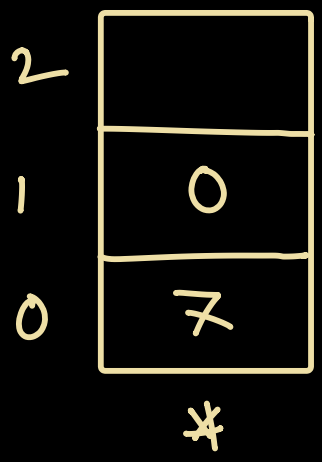
→ lower position গুলো আগে accomodate হবে।



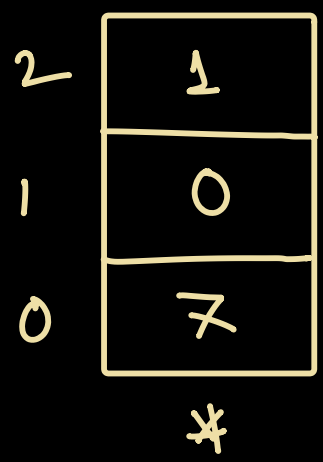
i.e: page 7 এর জন্য frame না থাকায়,



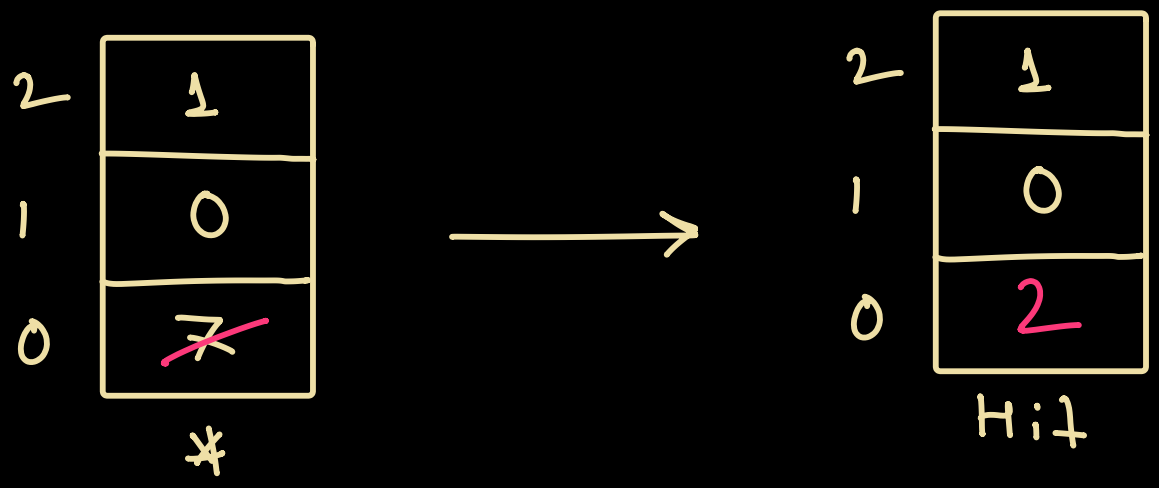
next page 0 এর জন্য page fault হবে,



next page 1 এর জন্য page fault হবে,

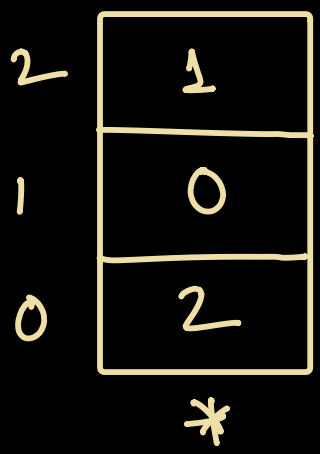


next page 2 এর জন্য page fault হবে,

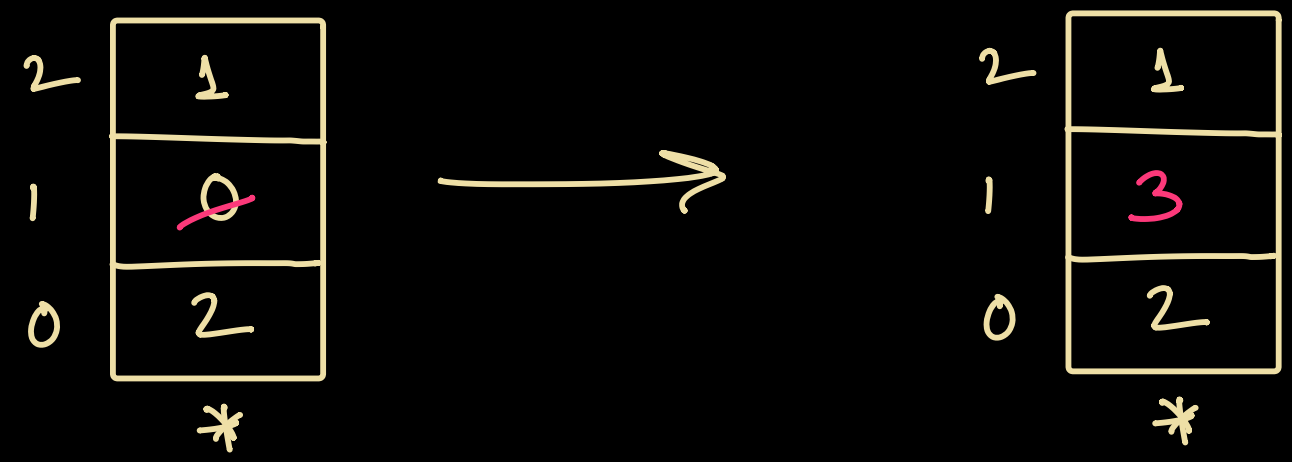


এখানে আগের 7 এর entry হচ্ছে। so replace with 7.

next page 0 এর জন্য page hit হবে। so no replacement.



next page 3 এর জন্য page fault হবে।



next page 0 এর জন্য page fault হবে

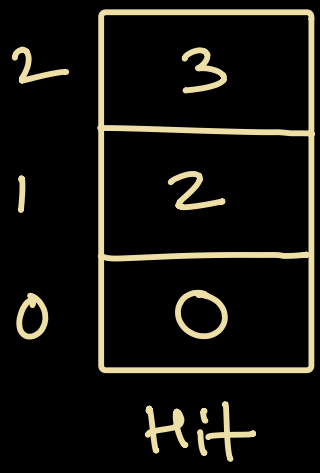
next page 9 এর জন্য page fault হবে

next page 2 এর জন্য page fault হবে

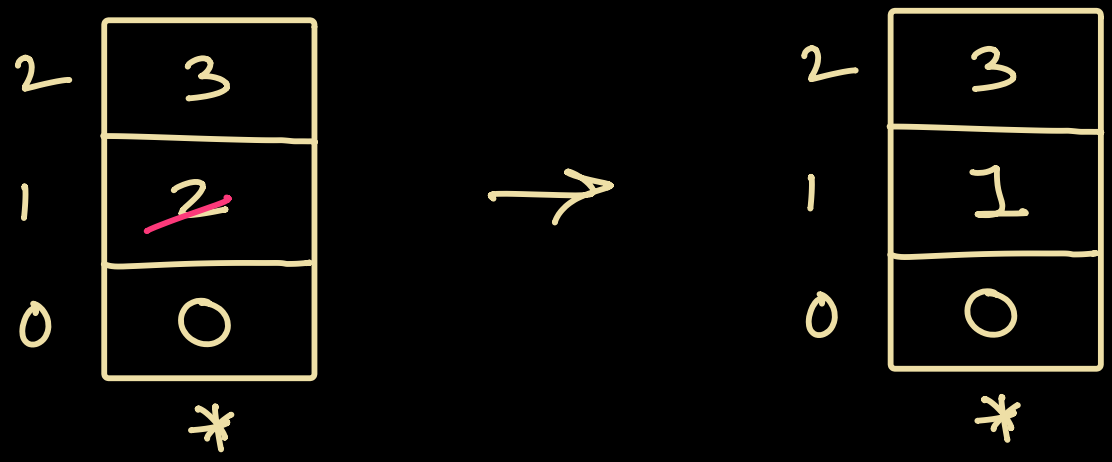
next page 3 এর জন্য page fault হবে

next page 0 এর জন্য page fault হবে

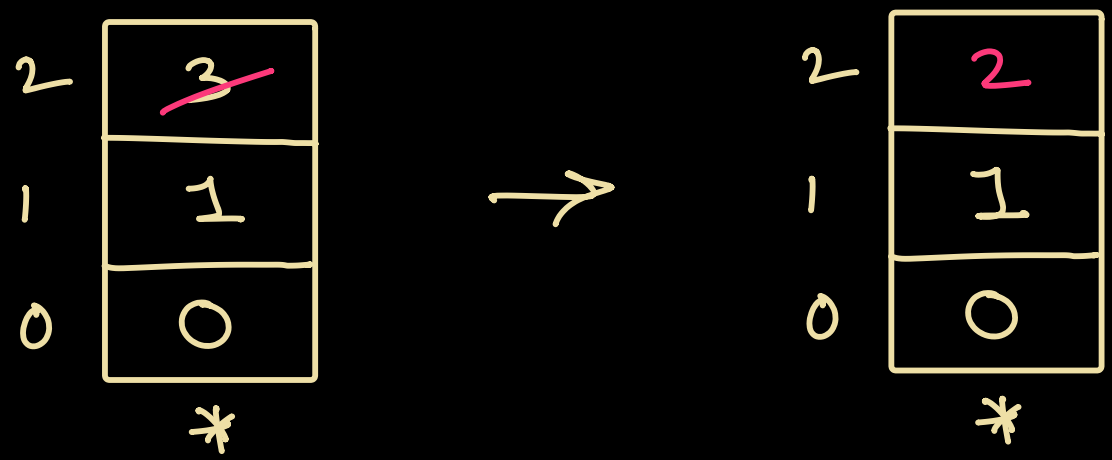
next page 3 এর জন্য page hit হবে



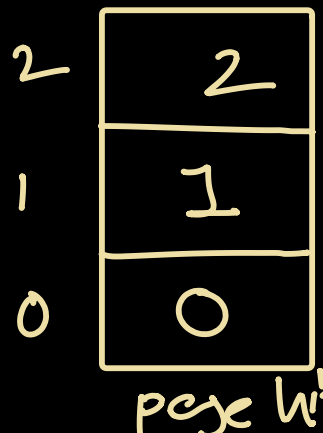
next page 1 এর জন্য page fault হবে:



next page 2 এর জন্য page fault হবে:



next page 0 এর জন্য page hit



observation from the simulation:

i) # hit (= 3)

ii) # page fault (= 12)

∴ total page ছিল 15 টা, total table 15 টা।

$$\text{page hit ratio} = \frac{(\#\text{page hit}) * 100}{\#\text{reference strings}}$$

(i.e: ↳ $= \frac{3 \times 100}{15} = 20\%$)

$$\text{page fault ratio} = \frac{(\#\text{page fault}) * 100}{\#\text{referenca string}}$$

(

$$i.e. : = \frac{12 \times 100}{15} = 80\%$$

11) Optimal Page Replacement Algorithm:

→ (not the optimal solution necessarily, depends on the scenario)

→ page hit ratio ↑ algo তত ভালো

→ works on the basis: CPU কোন page future এ সবচেয়ে পরে demand করছে।

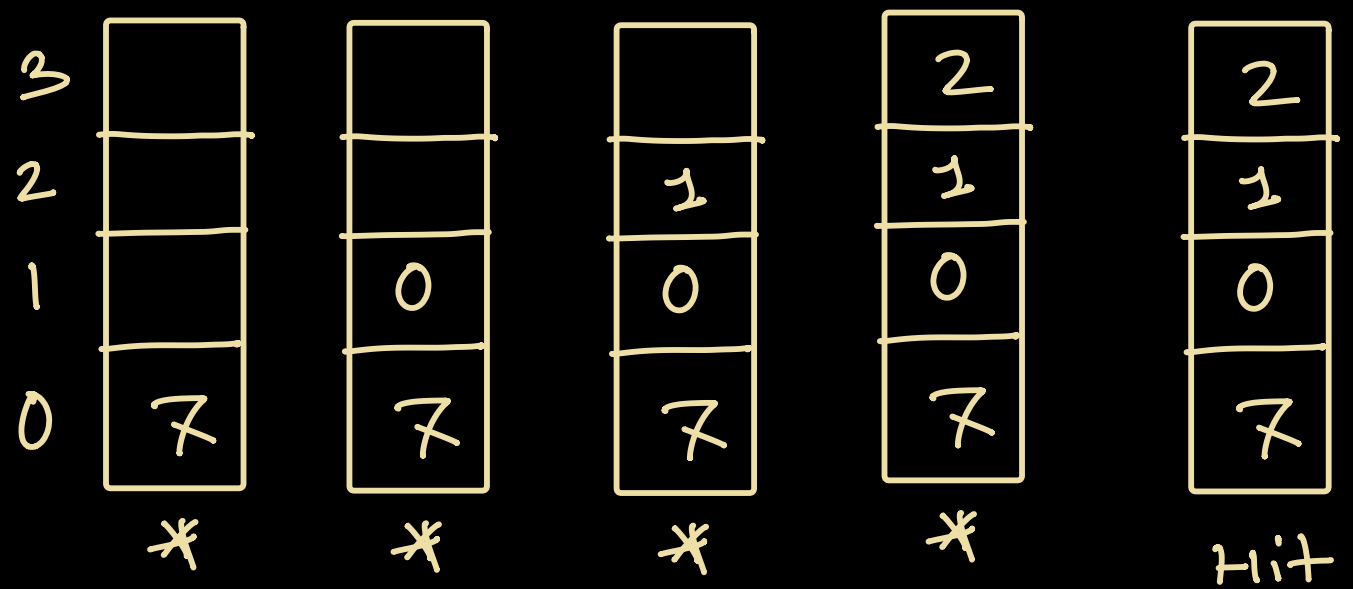
→ এমন page replace হবে যেটা future এ সবচেয়ে পরে demand করা হচ্ছে।

Que

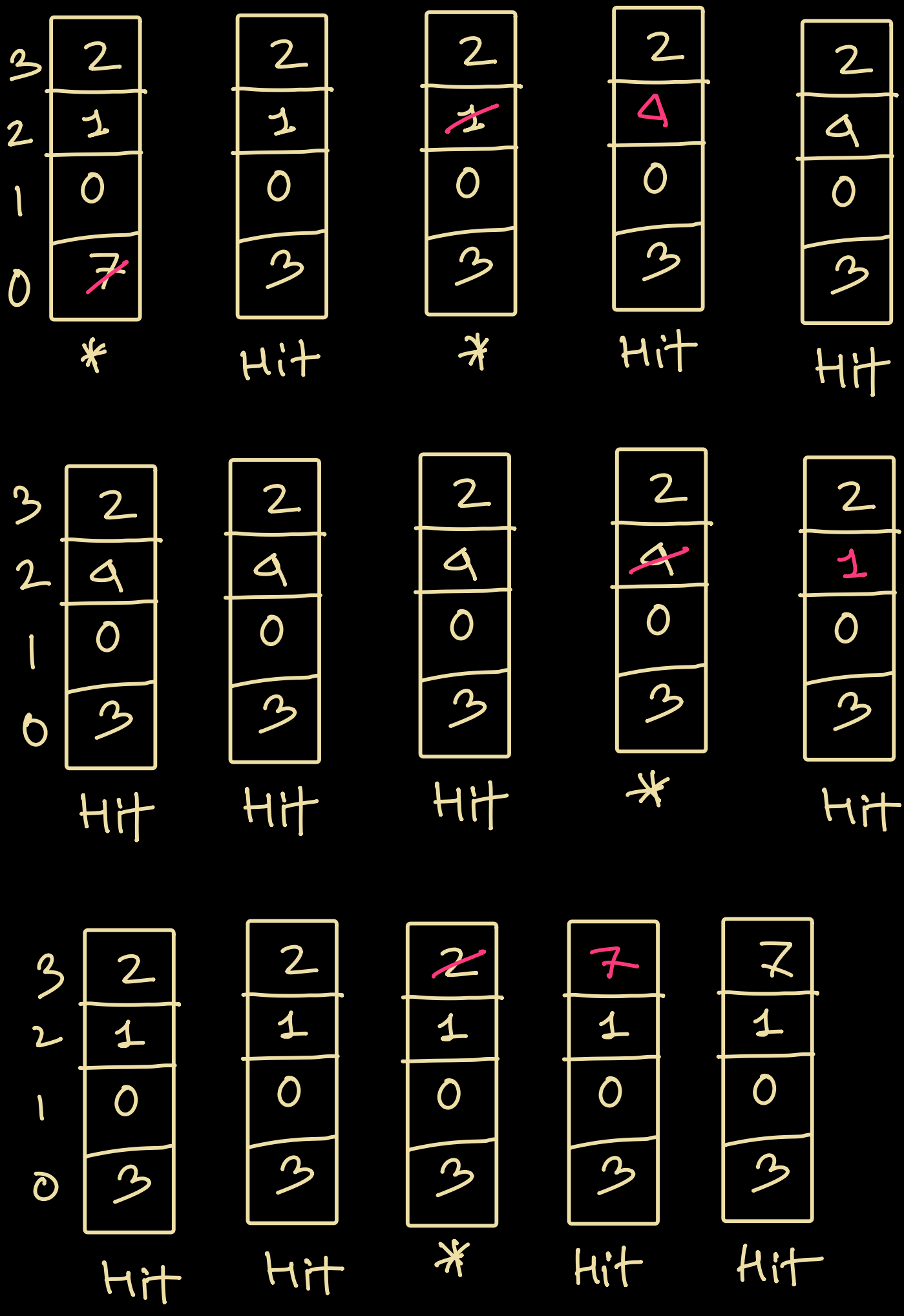
frames = 4

Ref string: 7, 0, 1, 2, 0, 3, 0, 4, 2, 3, 0, 3, 2, 1, 2, 0, 1, 7, 0, 1.

solve:



(next 7 was at furthest)


3 2 1 0
2 1 0 3
*
Hit
2 1 0 3
*
2 4 0 3
Hit
2 4 0 3
Hit
3 2 1 0
2 4 0 3
Hit
2 4 0 3
Hit
2 4 0 3
Hit
2 4 0 3
*
2 1 0 3
Hit
3 2 1 0
2 1 0 3
Hit
2 1 0 3
Hit
2 1 0 3
*
7 1 0 3
Hit
7 1 0 3
Hit

(# tables = # strings = 20)

Observations:

page hits = 12

page faults = 8

page hit ratio = $\frac{12 \times 100}{20}$

= 60%.

page fault ratio = $\frac{8 \times 100}{20}$

= 40%.

(iii) Least Recently Used (LRU) Page Replacement Algorithm:

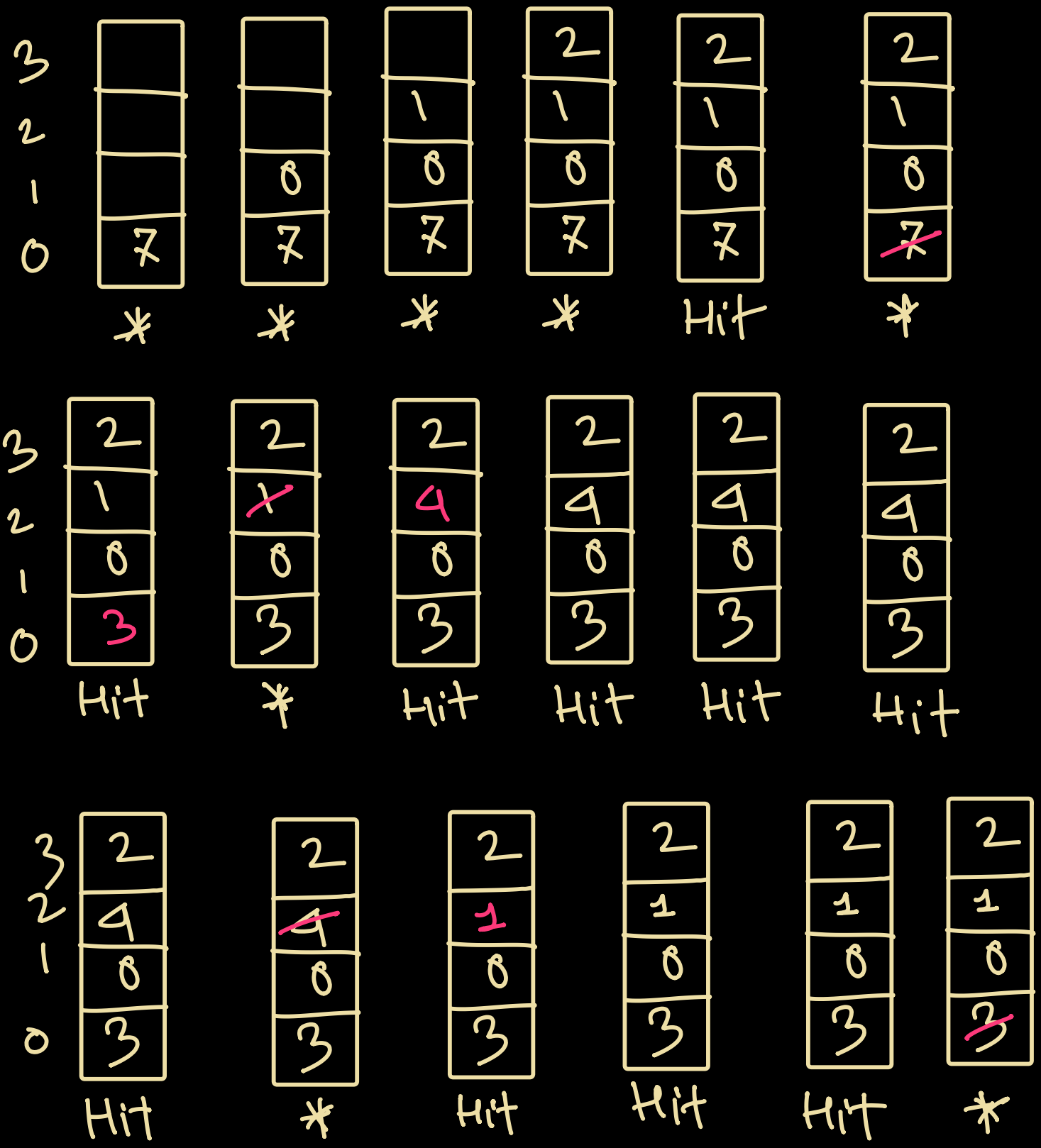
→ replace with the page past a সবচেয়ে আগে demand করা হয়েছিল।

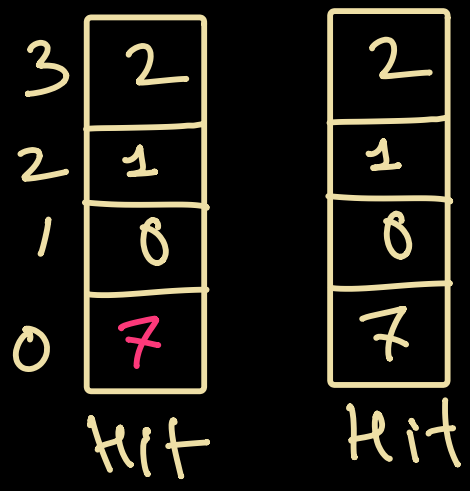
Que

frames = 4

Ref string: 7, 0, 1, 2, 0, 3, 0, 4, 2, 3, 0, 3, 2, 1, 2, 0, 1, 7, 0, 1.

solution:


3
2
1
0
7
0
1
2
2
2
7
1
0
7
1
0
7
1
0
7
*
*
*
*
Hit
*
3
2
1
0
2
1
0
3
2
x
0
3
2
4
0
3
2
4
0
3
2
4
0
3
Hit
*
Hit
Hit
Hit
Hit
3
2
1
0
2
4
0
3
2
4
0
3
2
1
0
3
2
1
0
3
2
1
0
3
2
1
0
3
Hit
*
Hit
Hit
Hit
*



page hit = 12

page fault = 8

$$\text{page hit ratio} = \frac{12 \times 100}{20}$$

$$= 60\%.$$

$$\text{page fault ratio} = \frac{8 \times 100}{20} = 40\%.$$