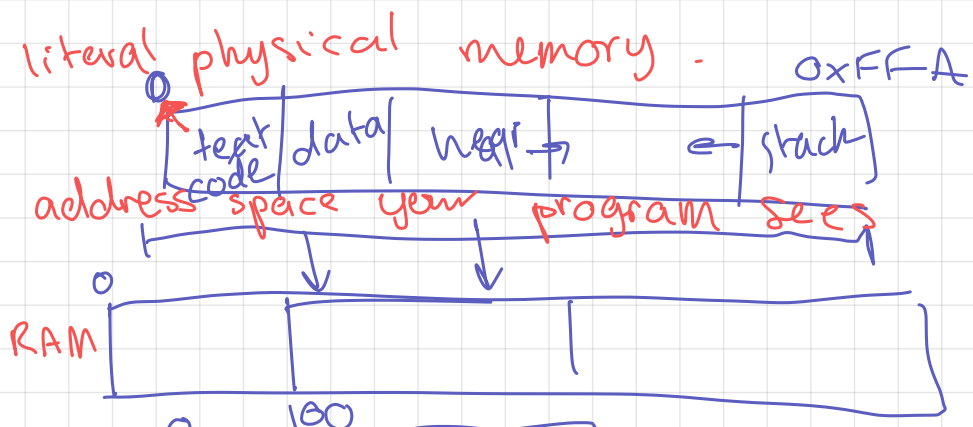
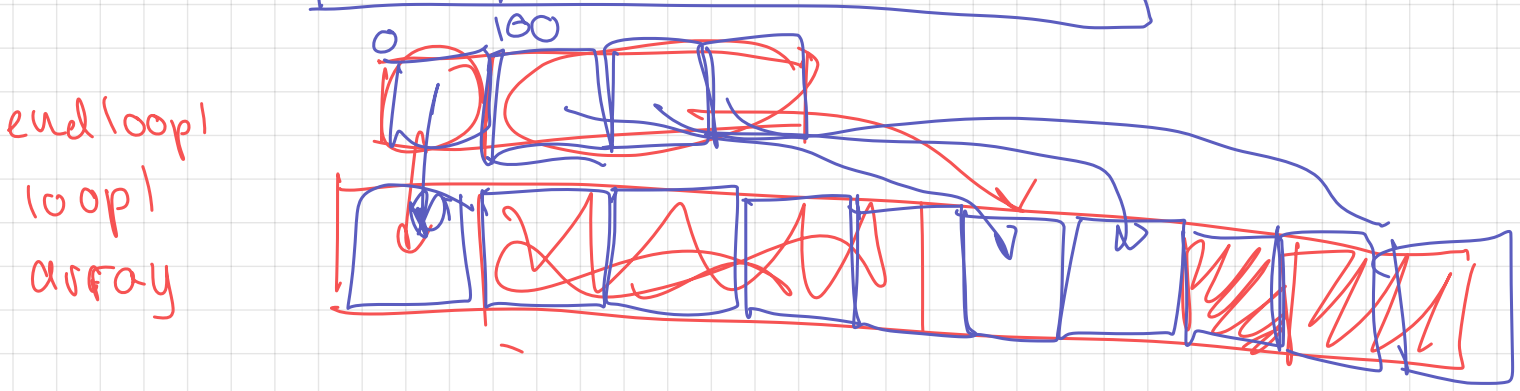
Week 10

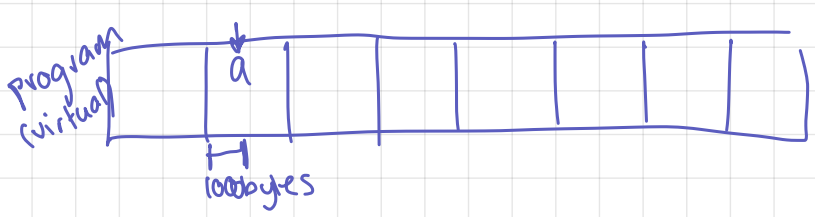
(7) physical - literal physical memory - 0xFFFA



6)



8) a) virtual address = 5096



virtual page number = $\left\lfloor \frac{\text{virtual address}}{\text{page size}} \right\rfloor$ = 1

∴ physical page # = 1

page_offset = virtual_address - virtual_page_# x page_size

= 5096 - 1 x 4096

= 1000

∴ physical_address = physical page # x pagesize + page_offset

= 1 x 4096 + 1000

= 5096

b) v-page-number = 6

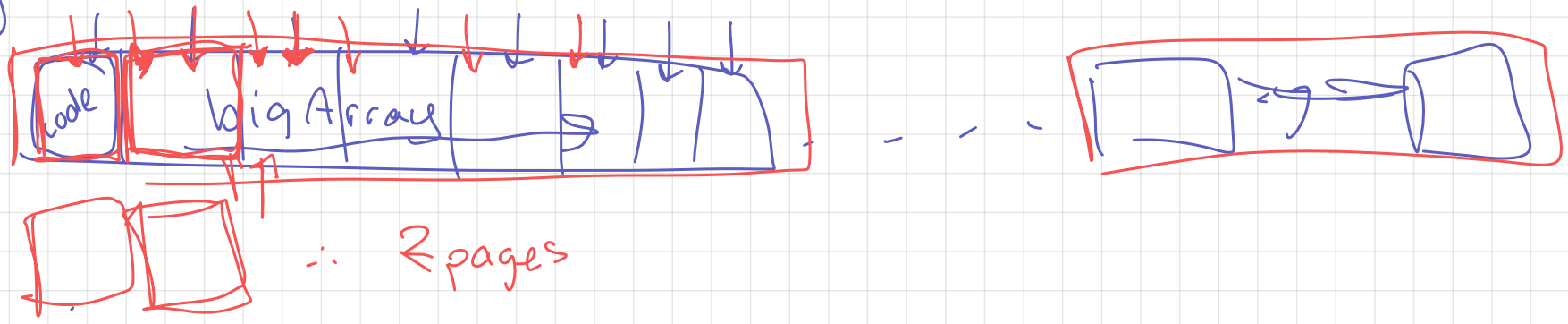
page-offset = 4092

physical-page = 2

physical_address = 12284

c) diy

9) a)



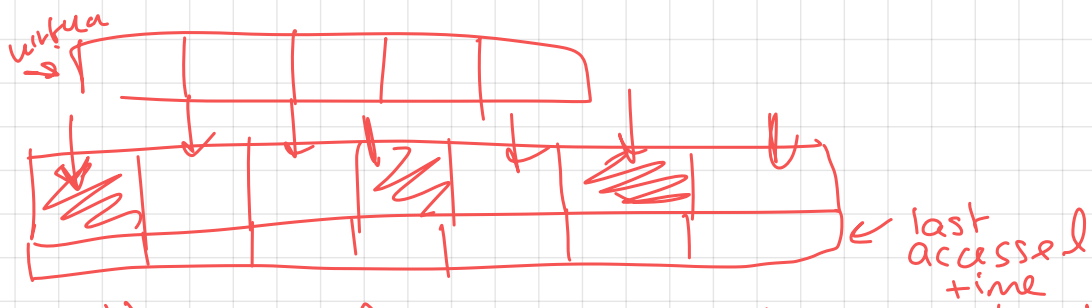
∴ 2 pages

b) 97 pages of array only accessed once

/only page fault @ start/first access

+ 1 page fault for first code access

98 total



1) see if page is already loaded

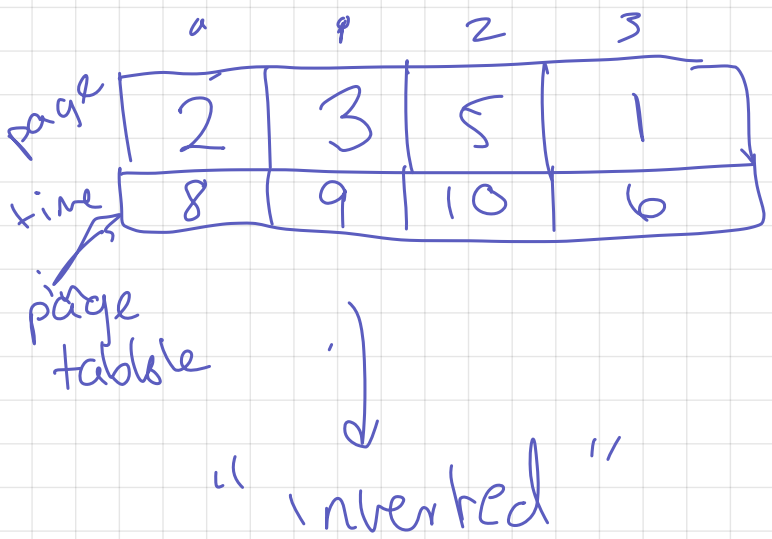
if is:
2a) update access time + use it

if not:
2b) find space to load into
a) look for empty page slot in physical memory

if empty slot:
1) load it in + store access time then use it

if not:
find something to overwrite
→ find the page that was least recently used
↳ then replace it.

3)



t=0 everything empty
t=1 5
t=2 3
t=3 5
t=4 3
t=5 0
t=6 1
t=7 2
t=8 2
t=9 3
t=10 5

10)

program's page

	[0]	[1]	[2]	[3]	[4]
t=0 status	not loaded	not loaded	not loaded	not loaded	not loaded
Frame No	–	–	–	–	–
Last Access	–	–	–	–	–

(initial values)

	[0]	[1]	[2]	[3]	[4]
t=1 status	read	not loaded	not loaded	not loaded	not loaded
Frame No	0	–	–	–	–
Last Access	1	–	–	–	–

read page 0

	[0]	[1]	[2]	[3]	[4]
t=2 status	read	not loaded	not loaded	not loaded	read
Frame No	0	–	–	–	1
Last Access	1	–	–	–	2

read page 4

	[0]	[1]	[2]	[3]	[4]
t=3 status	read	not loaded	not loaded	not loaded	read
Frame No	0	–	–	–	1
Last Access	3	–	–	–	2

read page 0

	[0]	[1]	[2]	[3]	[4]
t=4 status	read	not loaded	not loaded	not loaded	write
Frame No	0	–	–	–	1
Last Access	3	–	–	–	4

write page 4

	[0]	[1]	[2]	[3]	[4]
t=5 status	read	read	not loaded	not loaded	write
Frame No	0	2		–	1
Last Access	3	5		–	4

read page 1

	[0]	[1]	[2]	[3]	[4]
t=6 status	read	read	not loaded	read	write
Frame No	0	2		3	1
Last Access	3	5		6	4

read page 3

	[0]	[1]	[2]	[3]	[4]
t=7 status	not loaded	read	read	read	write
Frame No	→ 0	2	0	3	1
Last Access	– 3	5	7	6	4

read page 2

	[0]	[1]	[2]	[3]	[4]
t=8 status	not loaded	read	write	read	write
Frame No	→ 0	2	0	3	1
Last Access	– 3	5	8	6	4

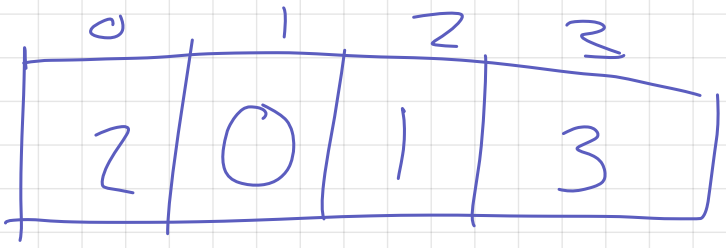
write page 2

	[0]	[1]	[2]	[3]	[4]
t=9 status	not loaded	read	write	read	write
Frame No	→ 0	2	0	3	1
Last Access	– 3	9	8	6	4

read page 1

	[0]	[1]	[2]	[3]	[4]
t=10 status	read	read	write	read	not loaded
Frame No	4	2	0	3	~
Last Access	10	9	8	6	~

read page 0



page table for each process

→ each column corresponds to a virtual page

status: how it was last used (or not loaded) if currently not loaded

"Frame" = "physical page"

→ Frame no = what physical page it's loaded into

Last Access = last access time.