Question 1)

Part a)

Logical address is generated by CPU whereas Physical address is located in Memory Unit.

Logical Address space is a set of all Logical addresses generated by CPU in reference to a program whereas Physical Address is set of all physical addresses mapped to the corresponding logical addresses.

Users can view logical addresses but cannot visit physical addresses.

Part b)

In Large pages our Page table is smaller than the case that our pages are Small

so finding the correspond page and after that corresponding frame is faster.

In Small pages our unused space is smaller than the case that our pages are Large

and we say in small pages we have Internal fragmentation whereas in Large pages we say we have external fragmentation.

So in overall in the trade of Time and Place we say Large pages are better for Time and worse for Place (more unused place) whereas Small Pages are better for Place (less unused place) and worse for Time (more time required to search).

Question 2)

1EAF = 0001 1110 1010 1111

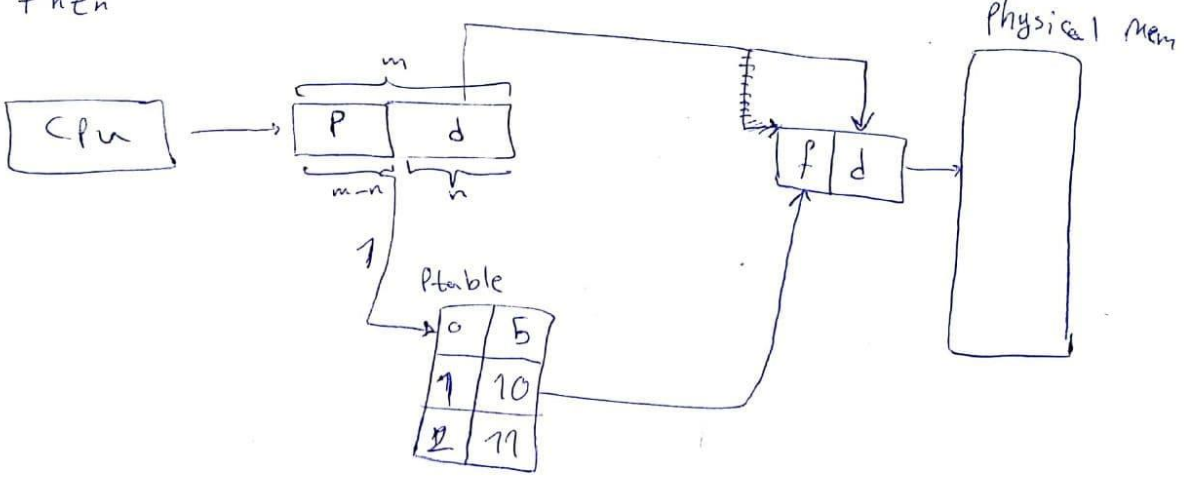
m-n = 4 bits, n = 12 bits

4KB = 4 × 2^{10} = 2^{12}

⇒ m = 16

n = 12

if we have standard logical to ~~exter~~ physical mapping of location then



So ⟶ 10 EAF ⟶ AEAF

Question 3)

32-bit logical address $\rightarrow 2^{32}$

$\downarrow$

Pointing to 2^{32} physical Page frames

Each Page size $\rightarrow$ 4KB $= 2^{12}$ bytes

Number of Pages in $Ptable_2$ $=$ $\frac{\text{Total Possible logical address entries}}{\text{Page size}}$

$= \frac{2^{32}}{2^{12}} = 2^{20} \approx 1$ million

Also as each entry in Page table is 4 bytes.
That means that the total size of the Page table in physical memory is $4 \times 2^{20} = 2^{22} =$ 4MB

Question 4)

Question (4)

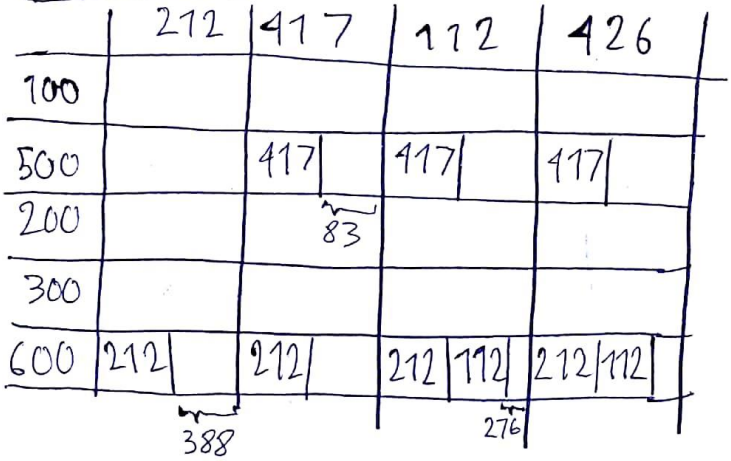
Mem Size → 100, 500, 200, 300, 600

Process Size → 212, 417, 112, 426

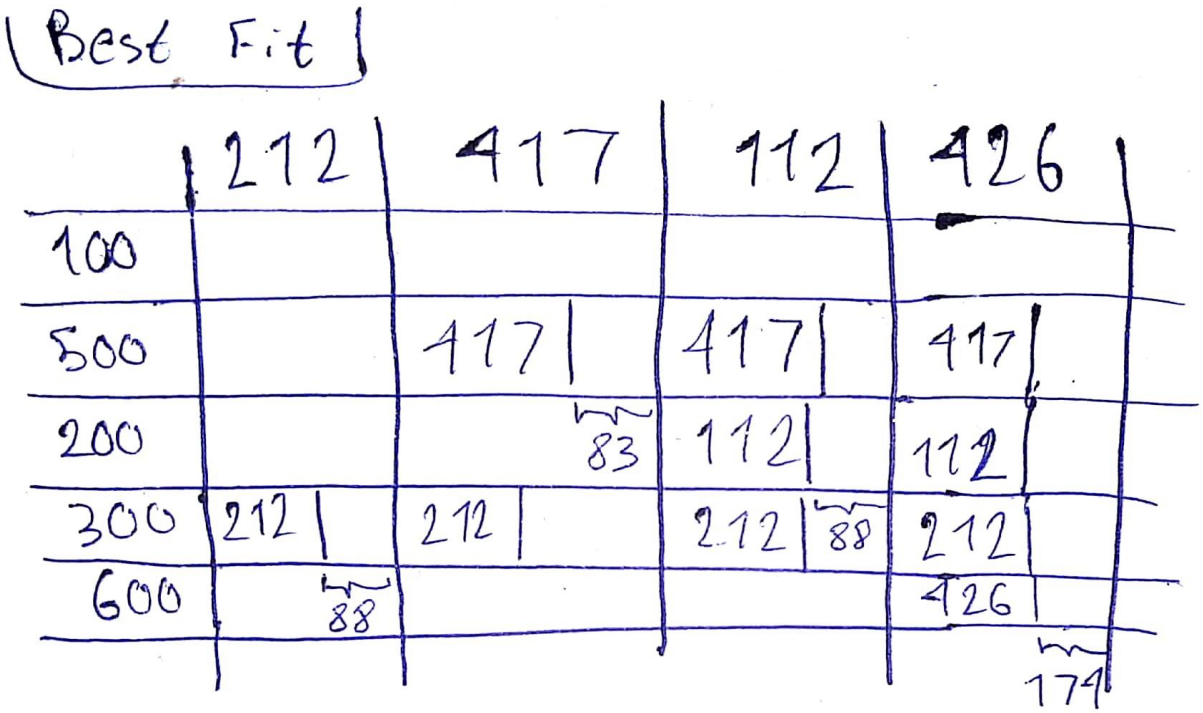
First Fit

So 426 is not allocatable in First Fit

Worst Fit



So 426 is not allocatable in WF



So in overall it is better to use BEST-FIT algorithm as if we use this algorithm in this scenario then we do not need to do a memory swap operation which is an overhead on the system.

It's true that first fit is faster then best and worst fit's but just like worst fit we need to do memory swap in order to allocate memory to the 4'th process but in best fit case we do not need any memory swap.

Question 5)

hit ratio → 80%

set associative time → 50 ns

Memory access time → 750 ns

(a)

~~EAT = 80/100 ×~~

With hit ratio:

search on set associative time + memory access time

50 + 750

= 800 ns

with miss ratio:

search on set ... + memory access time (for coping data from memory to TLB)

50 + 750

+ memory access time (to get the required data)

750

= 1550 ns

$$\Rightarrow EAT = \frac{80}{100} \times 800 + \frac{20}{100} \times 1550 = 950 \text{ ns}$$

(b) for n-level Paging the EAT is

$P\{TLB\ Access\text{-}time + Mem\ Access\ time\} + (1-P)(TLB\text{-}Access\ time + (n+1) \times Memory\ Access\ time)$

P = hit ratio
n = n-level Paging

for 2 level Paging

$$\frac{80}{100}(50+750) + \frac{20}{100}(50+3\times 750)$$

$$= 1100\ ns$$