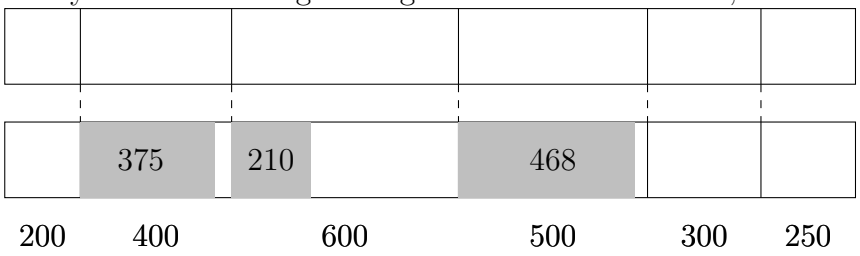
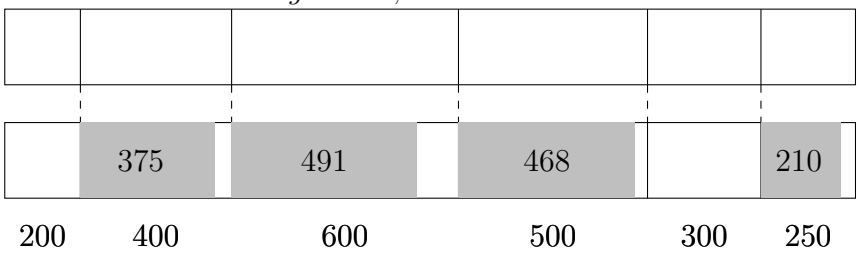
1. Assume that the minimum page size, such that the entire page table fits well in one page is x, we have $2^{28} \div x \times 4 = x$, thus $x = 2^{15}$. So the page size is 32KB.

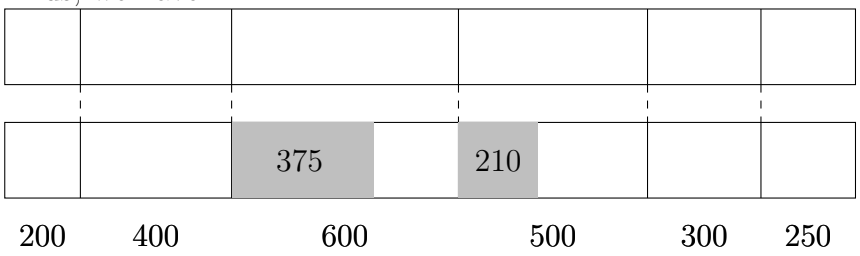
2. 1) If we use *First Fit Algorithm*, the fourth process have to wait until the system have a large enough hole to hold it. Thus, we have



   2) If we use *Best Fit Algorithm*, we have



   3) If we use *Worst Fit Algorithm*, the third and fourth processes need to wait until the system has large enough holes to hold them. Thus, we have



3. 1) Since the memory is byte addressable and virtual address is 64 bits long, so the virtual address space is 2^{64} bytes. One page can hold $2^{20}/2^2 = 2^{18}$ page table entries. If we use 2 levels of page table, then we can only cover $2^{18} \times 2^{18} \times 20^{20} = 2^{56}$ bytes of space, if we use 3 levels of page table, we will be able to cover $2^{18} \times 2^{18} \times 2^{18} \times 2^{20} = 2^{74}$ bytes. Thus, we need 3 levels of page table.

   2) The struct of virtual address is like below

1st index	2nd index	3rd index	page offset
8	18	18	20

The struct of physical address is like below

phycial frame number	page offset
20	

4. 1) The total number of page faults is 6, hit ratio is 40% and miss ratio is 60%.
   2) The total number of page faults is 6, hit ratio is 40% and miss ratio is 60%.
   3) The total number of page faults is 5, hit ratio is 50% and miss ratio is 50%.

5. $EAT = (100 + 20) \times 0.8 + (200 + 20) \times 0.2 = 140$ns.

6. - FCFS:

     $(100-23)+(89-23)+(132-89)+(132-42)+(187-42) = 421.$

   - SSTF:

     $(100-89)+(132-89)+(187-132)+(187-42)+(42-23) = 273.$

   - SCAN:

     $(100-89)+(89-42)+(42-23)+(23-0)+(132-0)+(187-132) = 287.$

   - C-SCAN:

     $(100-89)+(89-42)+(42-23)+(23-0)+(199-0)+(199-187)+(187-132) = 366.$

   - C-LOOK:

     $(100-89)+(89-42)+(42-23)+(187-0)+(187-132) = 319.$