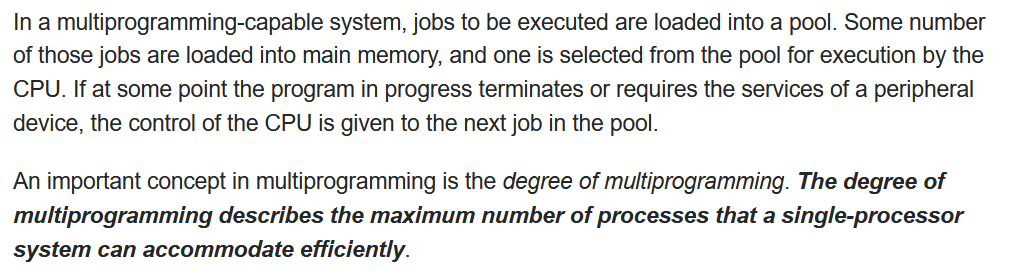
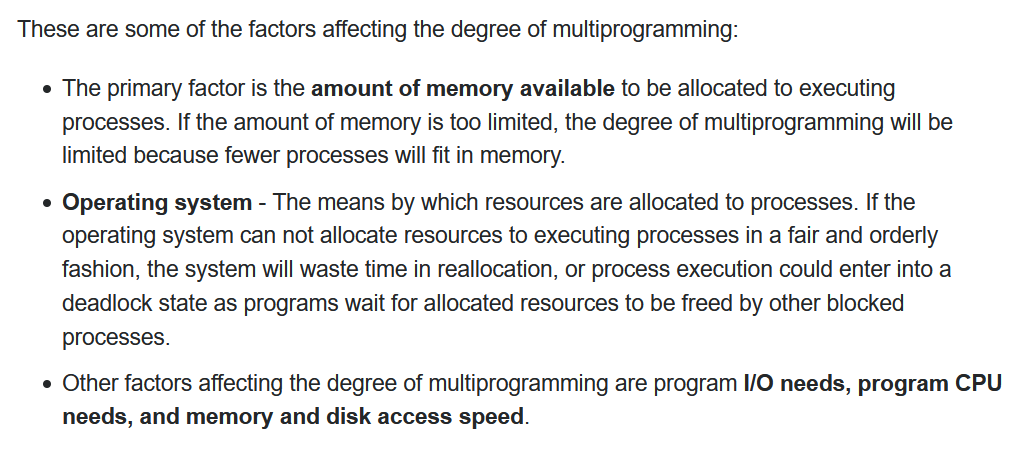


Define Degree of Multi-programming

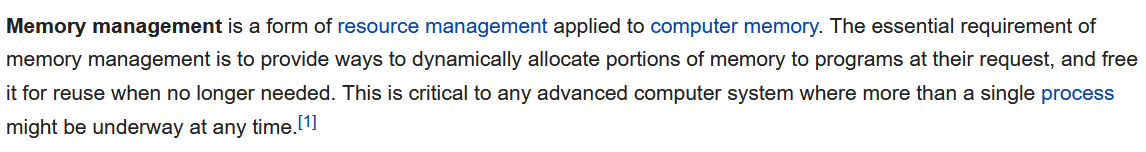




* More processes should be in RAM, so that the CPU utilization can be maximum.
* When the size of Memory increases, the number of process accommodating the RAM increases thereby increasing the CPU utilization

**Memory Management Technique**

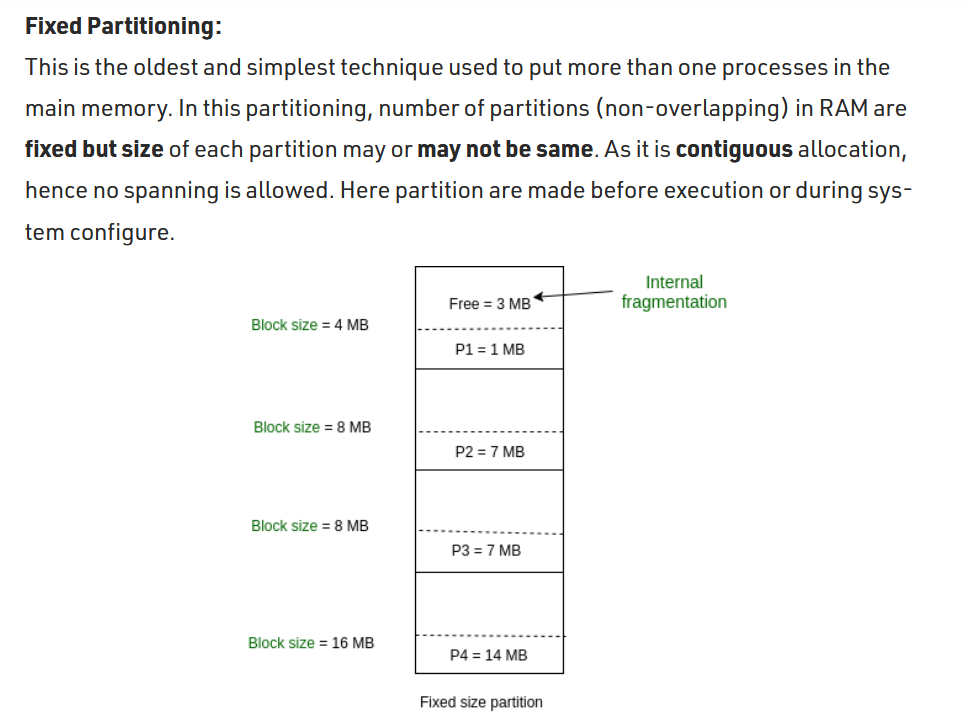
It is the technique of utilizing the RAM more efficiently.

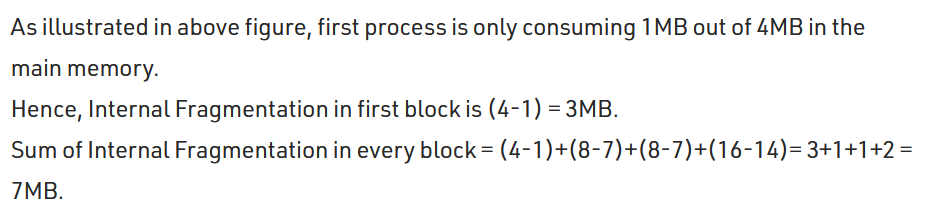
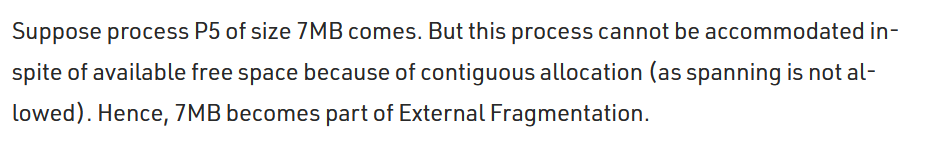


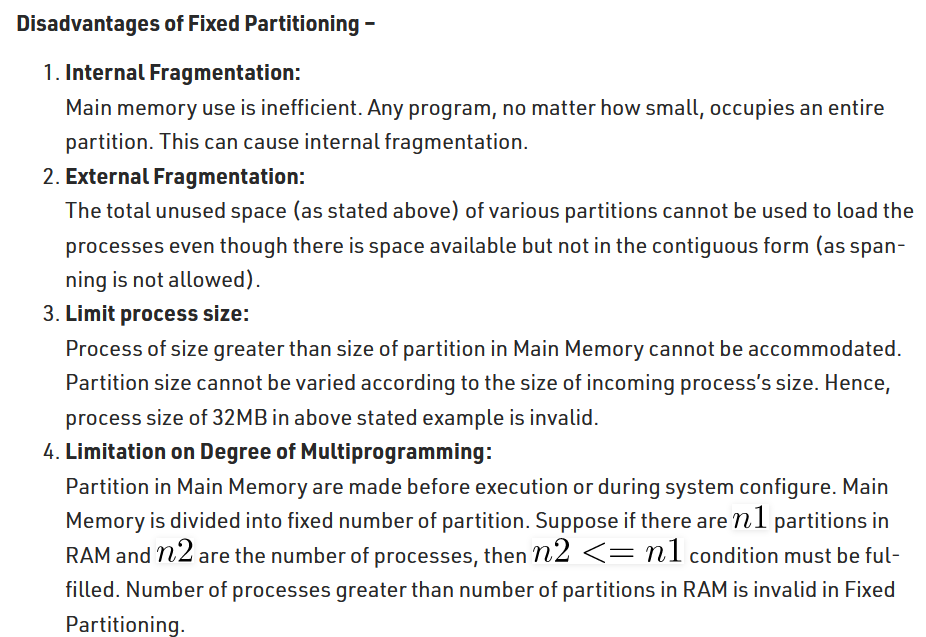
**Static/Fixed Partition**

When-ever the process comes into the RAM, before itself without considering the process size fixed spaces are allocated.

Number of partitions are fixed size but the size of each partition can be same/different size.  
No spanning is allowed.



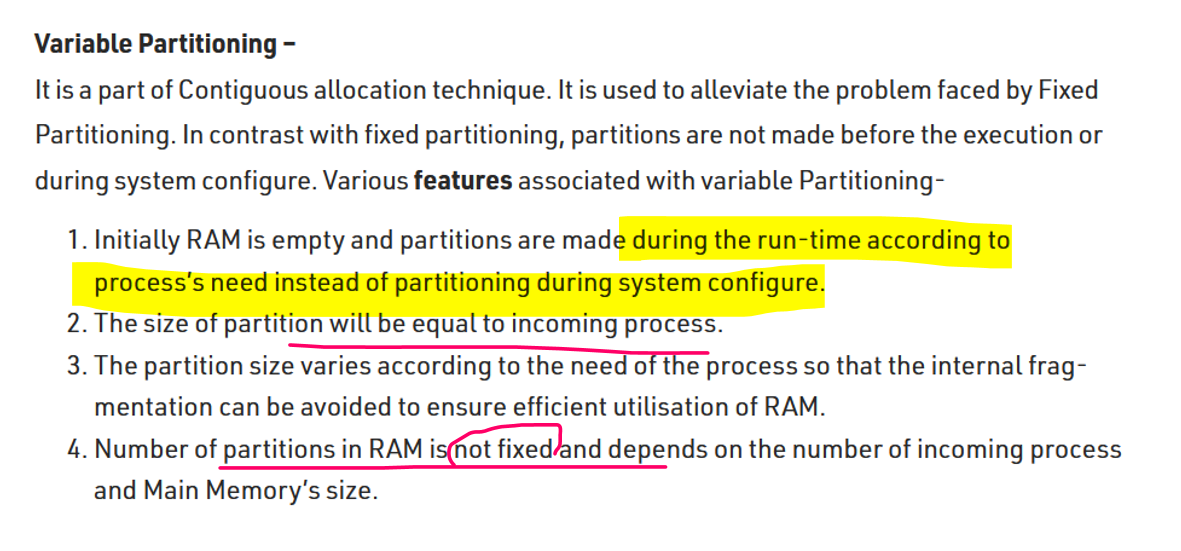


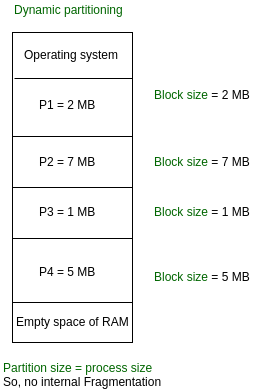
***Internal Fragmentation 🡪 The difference between the memory allotted and size of the process.***

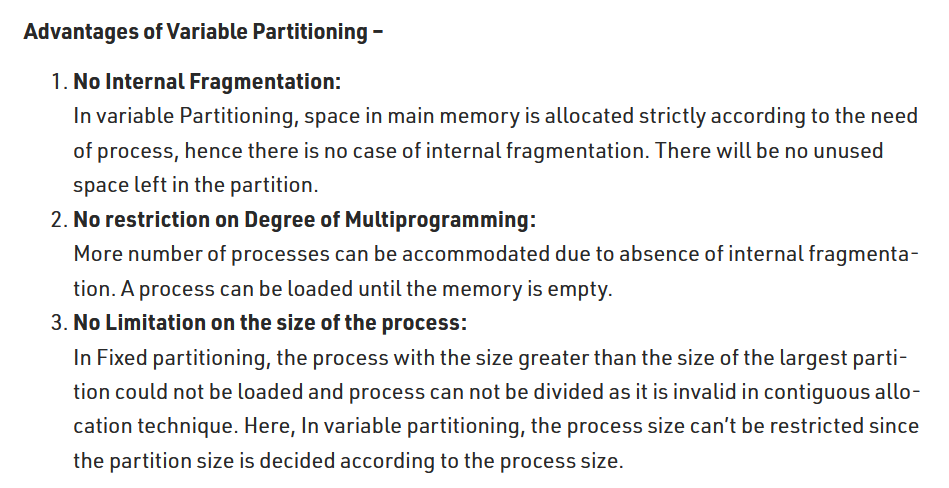
***External Fragmentation 🡪 The number of unused spaces in the memory can be grouped together, so as to accommodate one more process.***   
Due to contiguous memory allocation external fragmentation cannot be achieved.

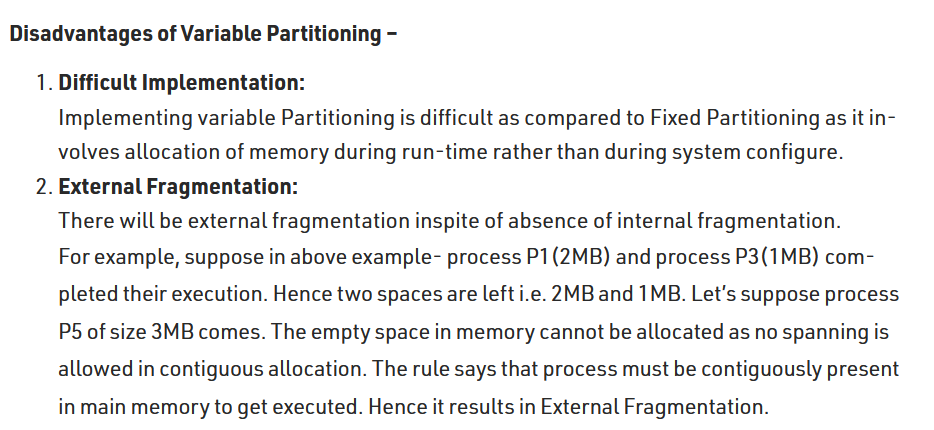
**Dynamic/Variable Partition**

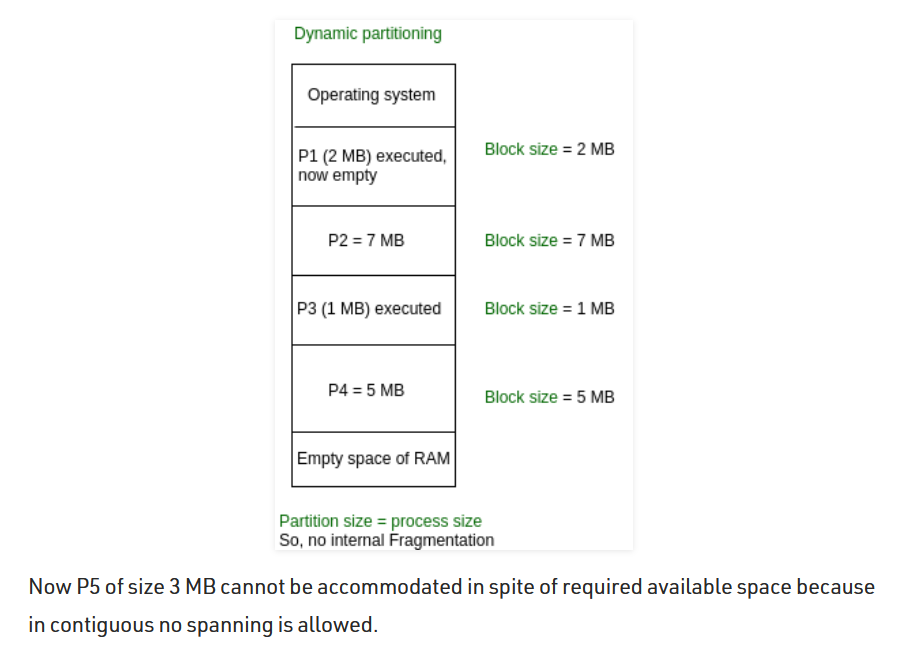
When-ever the process comes into the RAM, that time only the space is allocated to that process.





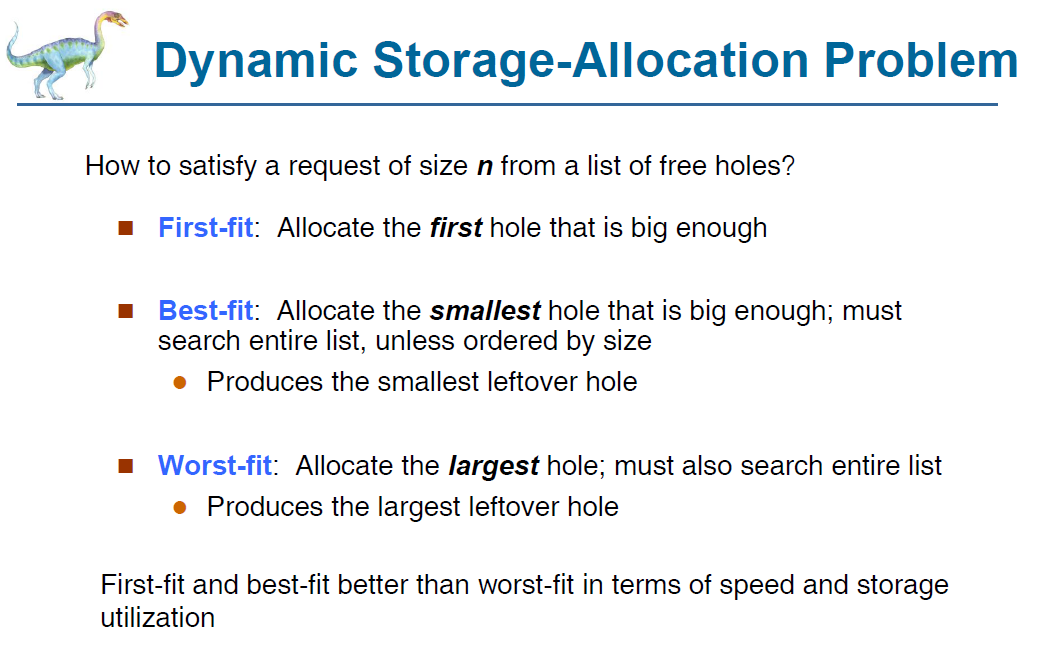




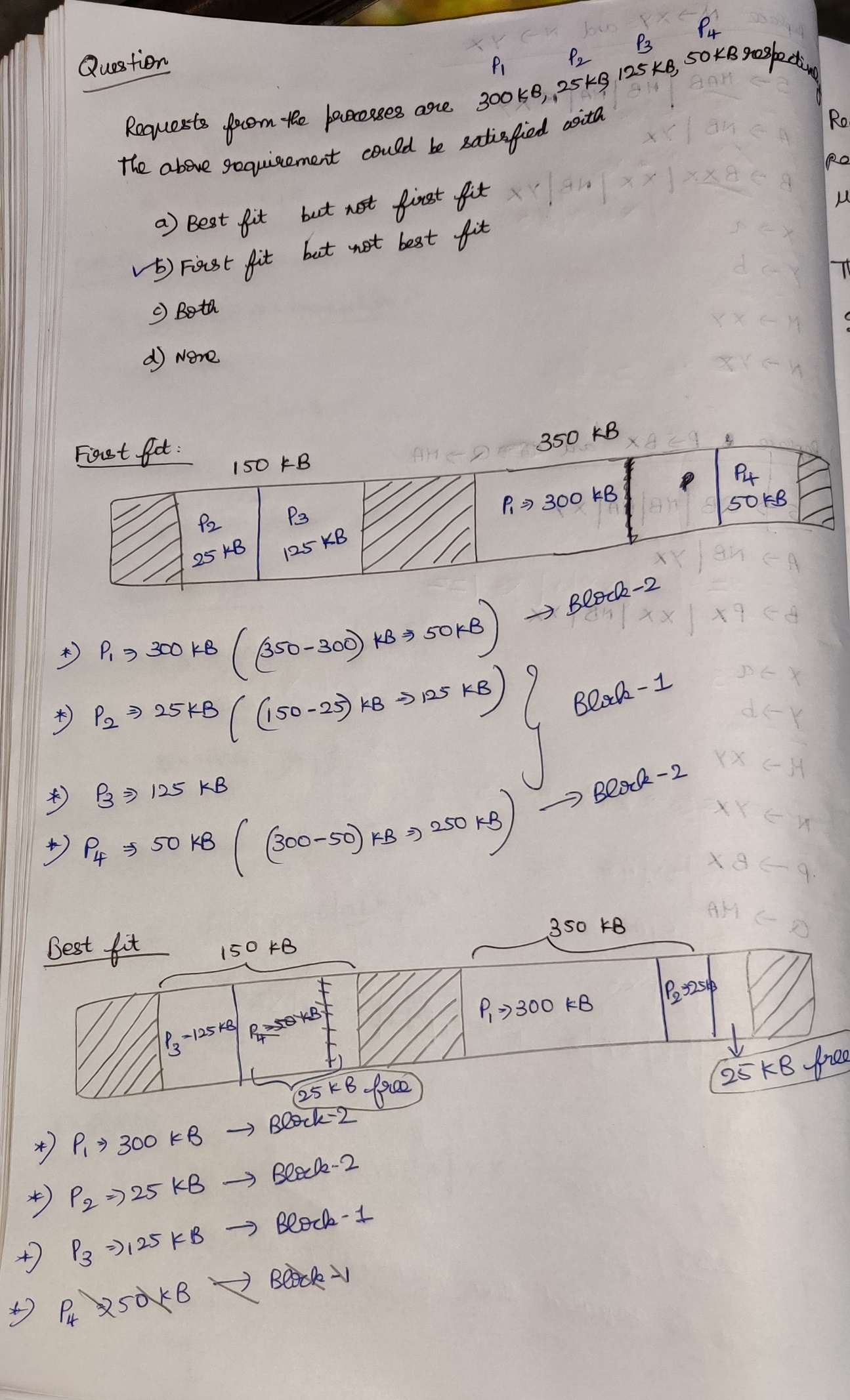


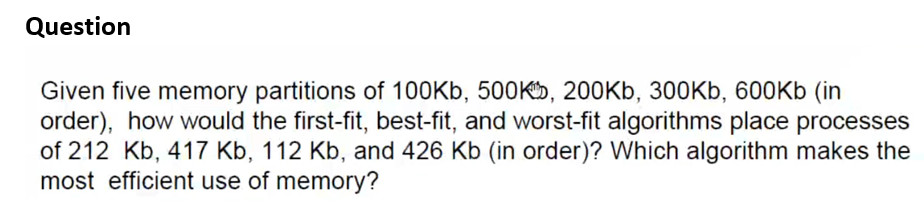
Although via external fragmentation the space is available due to continuous memory allocation, the left over space cannot be used.

Compaction is a method to remove external fragmentation.  
<https://www.javatpoint.com/os-compaction>   
Allocation and deallocation of holes and process’s are difficult.



**First fit 🡪** When we scan the RAM, we can find many holes. Among that the first hole is taken into consideration. That first hole can be large/small.  
P1 comes and resides, While P1 residing there may be some small space left out or some large space left out 🡪 Uncertain for the other process to reside in the left over place. **Speed 🡪 Very fast  
Best fit 🡪** Will compare all the holes in the RAM, it will look for the best match which is large enough to accommodate that process and that ***hole should lead to minimum internal fragmentation.  
The space after the process accommodates should be less.***  
It will scan and find-out the best match **Speed 🡪 Very low   
Worst fit 🡪**  Will look for the largest hole to fit the process.  
P1 comes in and resides. Mean-while since worst fit has the largest space for P1 after P1 resides also there are some good enough space for other process to reside.***The holes should lead to maximum internal fragmentation.***  
**Speed 🡪 Very low**

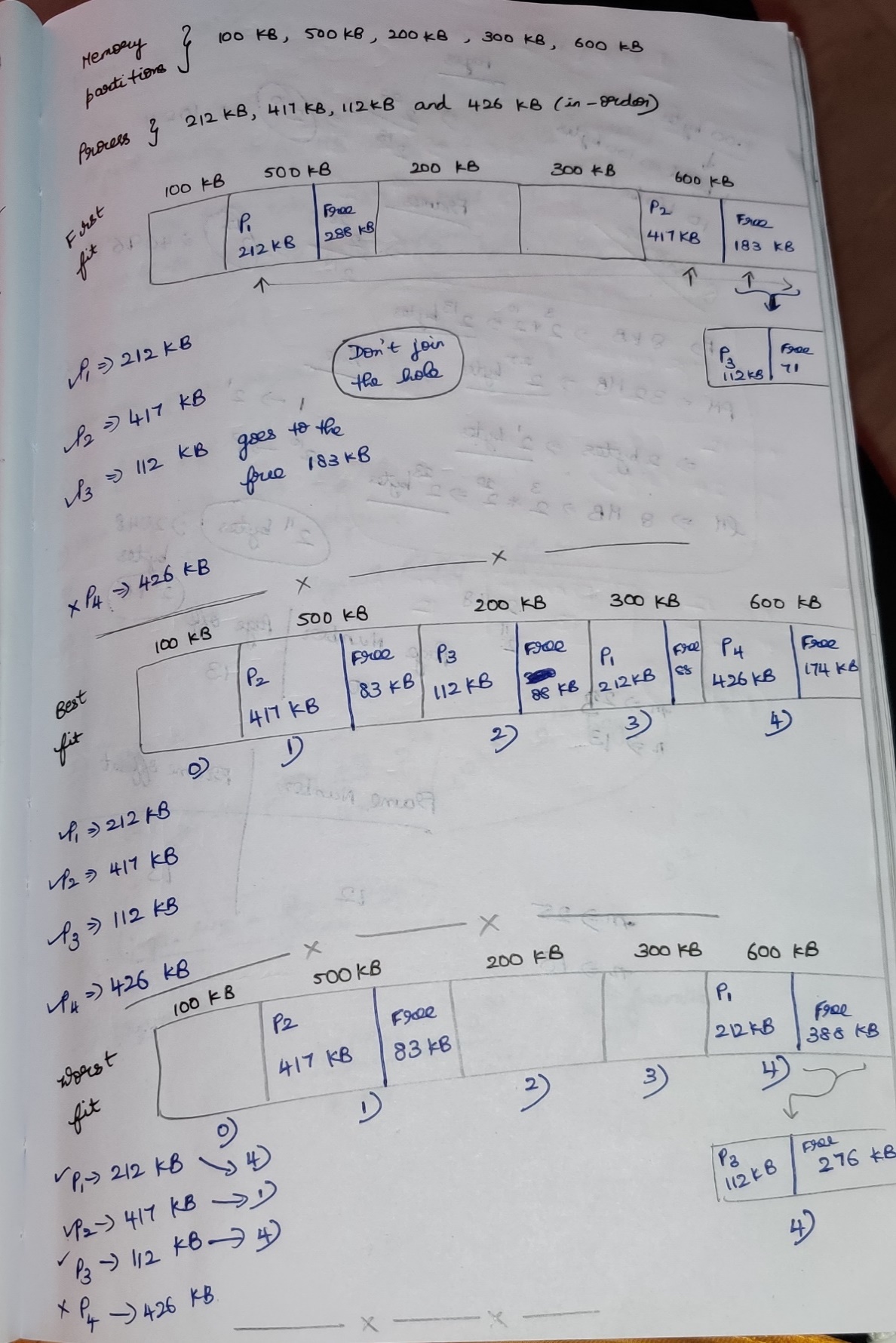




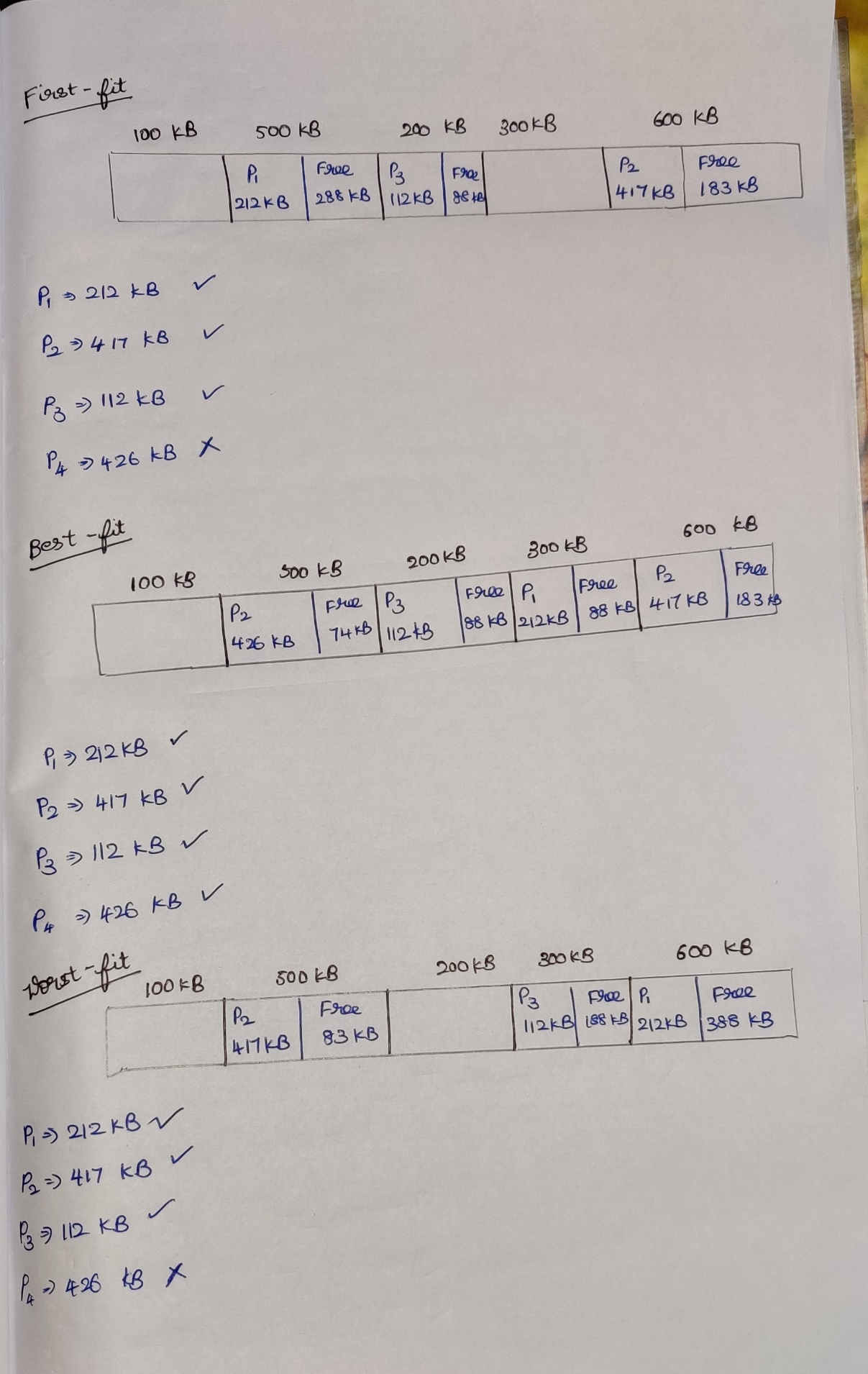
In first fit and in worst fit, P3 🡪 426KB was not having the space to enter into RAM, even-though as a whole there are some enough spaces to occupy P4 due to the partition of RAM P4 cannot able to occupy. 🡪 Fragmentation

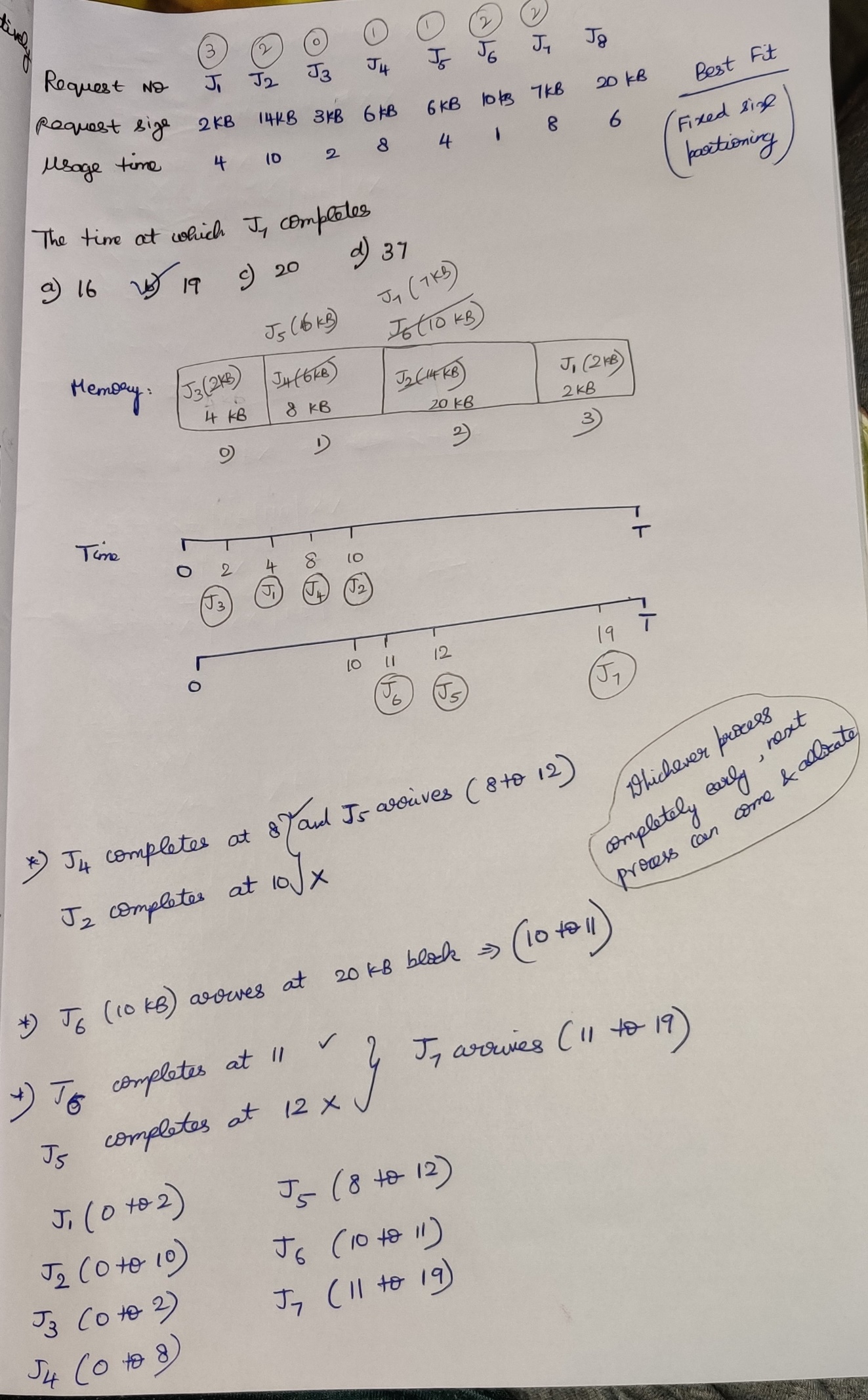
**Variable-size Partition**

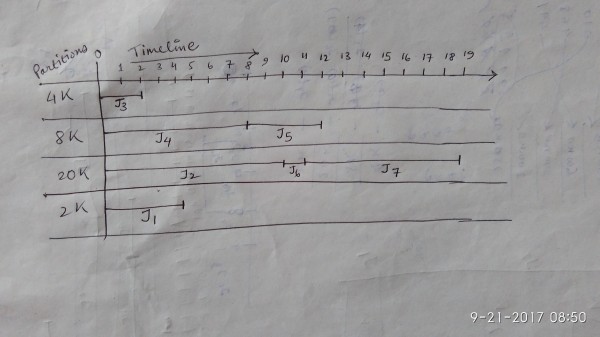
***(The remaining space of the block after allocating a process can be reused once again)***



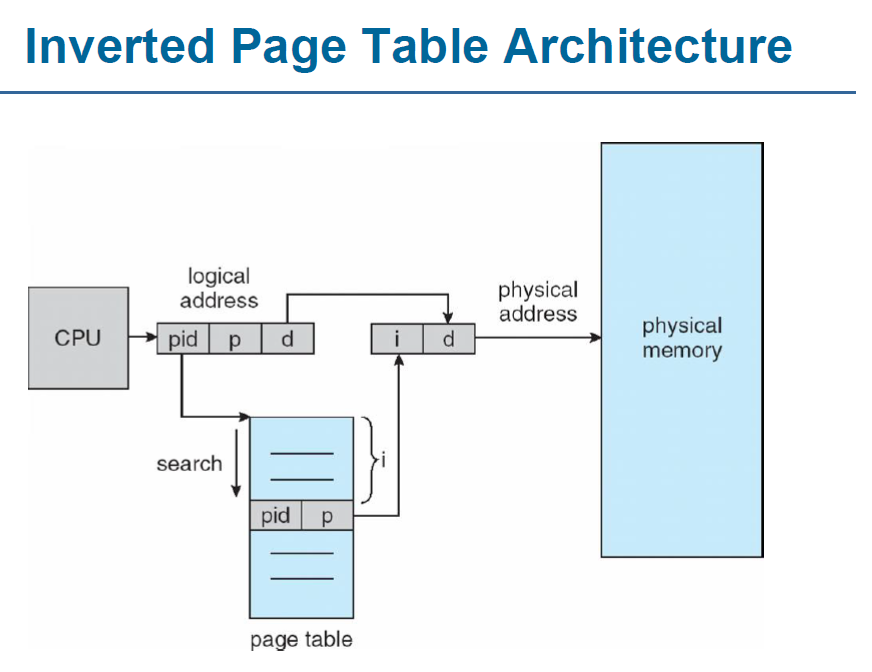
**Fixed-size Partitioning**

***(The remaining space of the block after allocating a process cannot be reused once again)***





**Inverted Paging**





**Question**Consider a virtual address space of 32 bits and page size of 4KB and the system is having a RAM of 128 KB. Then what is the ratio of size of page table to inverted page table if each entry in both is of size 4B???

***Logical Address space = Virtual Address space***

