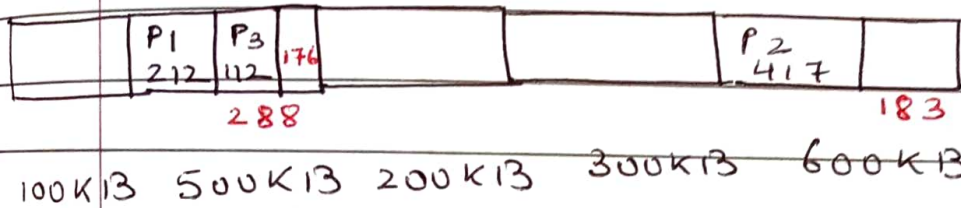## Numerical

1) Given memory partitions of 100 KB, 500 KB, 200 KB, 300 KB & 600 KB (in order), how would each of the first - fit, best - fit & worst - fit algorithms place processes of size 212 KB, 417 KB, 112 KB & 426 KB (in that order)? Which algo. makes the most efficient use of memory?

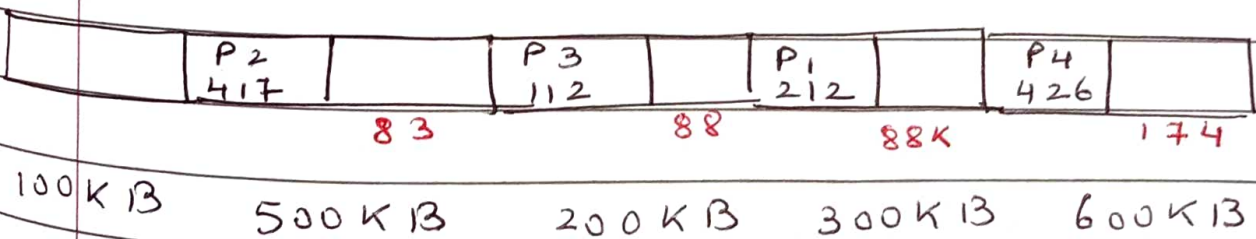Partitions :- 100KB, 500KB, 200KB,
300KB, 600KB

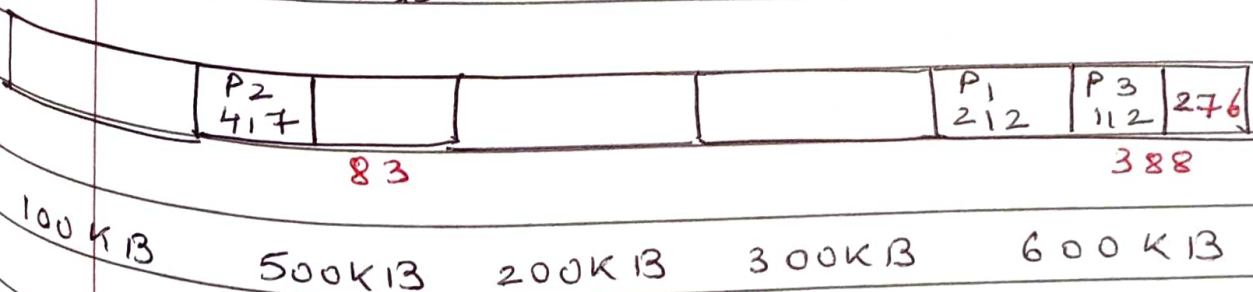Processes :- 212KB, 417KB, 112KB, 426KB
P_1 P_2 P_3 P_4

→ First Fit

	P1 212	P3 112	176			P2 417		

288

183

100KB 500KB 200KB 300KB 600KB

$P_4(426KB)$ cannot be added.

Best Fit

	P2 417			P3 112		P1 212		P4 426	

83 88 88K 174

100KB 500KB 200KB 300KB 600KB

Worst Fit

	P2 417					P1 212	P3 112	276

83 388

100KB 500KB 200KB 300KB 600KB

$P_4(426KB)$ cannot be added.

∴ Efficient algorithm is best fit.

2) Partitions 150K, 350K

Processes 300K, 25K, 125K, 50K
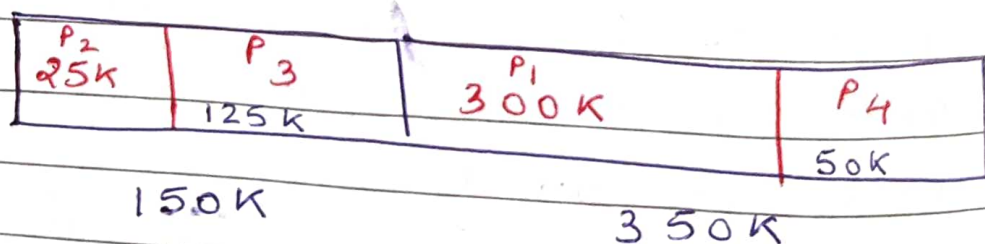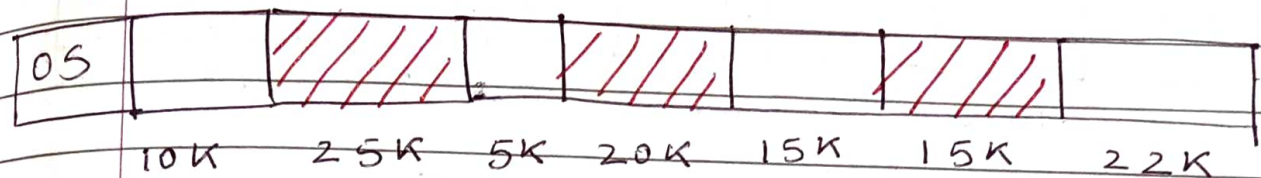P_1 P_2 P_3 P_4

→ First Fit

25K P_2	125K (P_3)	300 K (P_1)	50K (P_4)

150K   350K

Best Fit

125k	2 175K	300K (P_1)	50K (P_2)	25K

150K   350 K
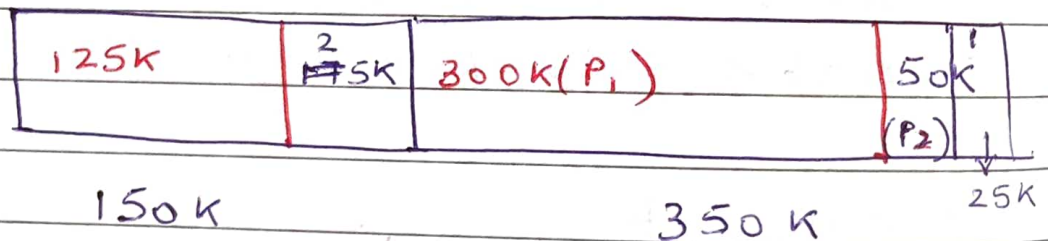
50K(P_4) cannot be added. (External fragmentation)

Worst Fit

P_2 25K	P_3 125K	P_1 300K	P_4 50K

150K   350K

First Fit & Worst Fit are efficient algorithms.

* **First Fit, Best Fit, Worst Fit Numerical**

OS		///		///		///	
	10K	25K	5K	20K	15K	15K	22K

Process Allocate memory for additional request of 4k & 10k. Compare the memory allocation using first fit, best fit, worst fit allocation methods in terms of external fragmentat⁓.
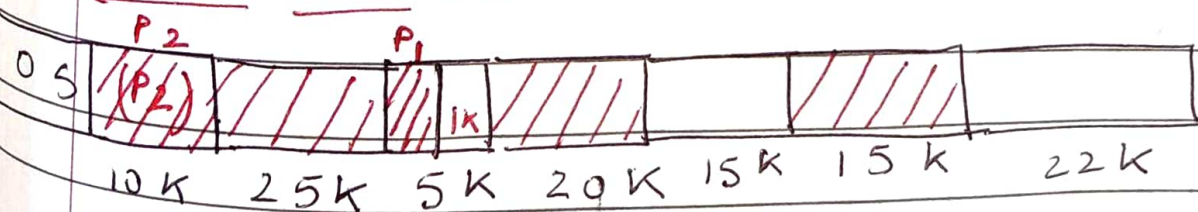
→ Solution

$$P_1 = 4K, \quad P_2 = 10K$$

OS	P₁				P₂		
	///	///		///	///	///	
	10K 6k	25K	5K	20K	15K 5k	15K	22K
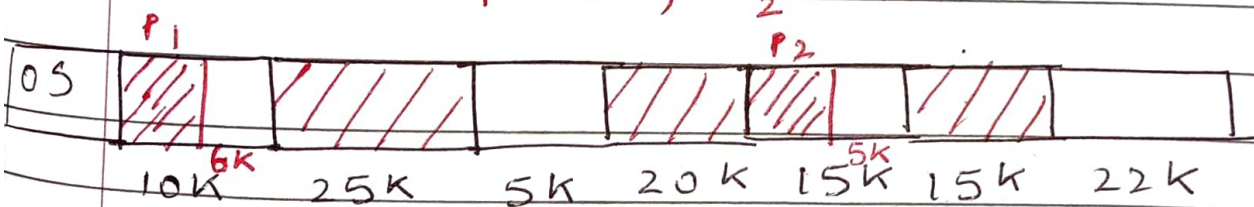
It leaves fragmentation of 6k + 5k = 11k.

Best Fit

OS	P₂		P₁				
	(P₂) ///	///	/// 1k ///		///		
	10K	25k	5K	20K	15K	15 k	22 k

It leaves fragmentat⁓ of 0k + 1k = 1k

Worst Fit

OS		10K	25K	5K	20K	15K	15K	22K

P1 4K | P2 10K | 8K
18K
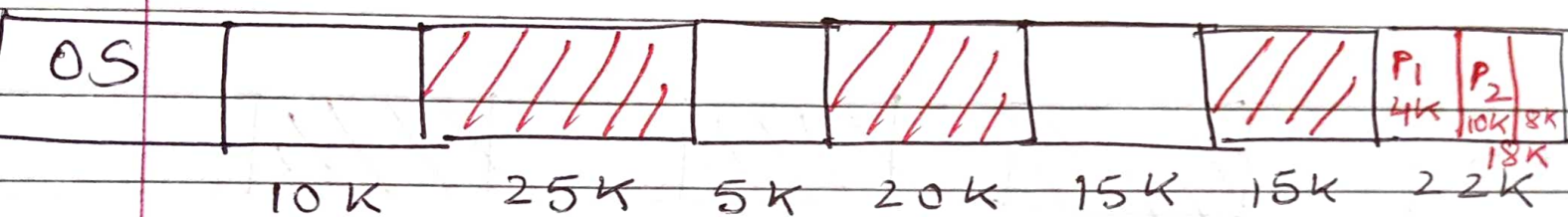
It leaves fragmentat of 8K

Best method is Best fit