Name: Nimra Nasir
Reg No: 19-CP-35

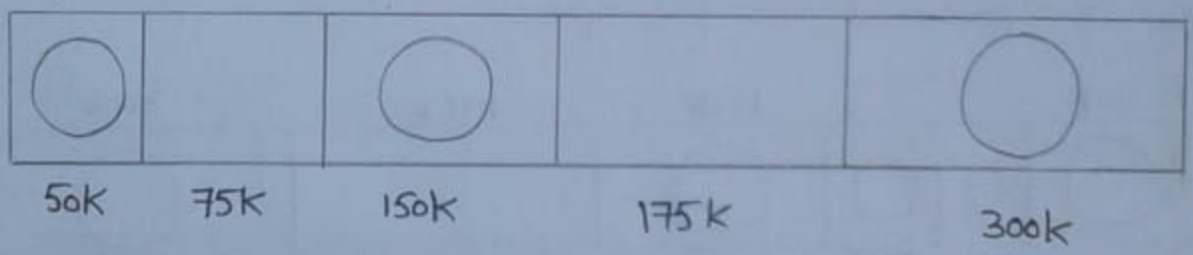
Operating System

Lab Manual: 9

Questions:

Question #1:

Process requests are given as;
25K, 50K, 100K, 75K



Determine the algorithm which can optimally satisfy this requirements.

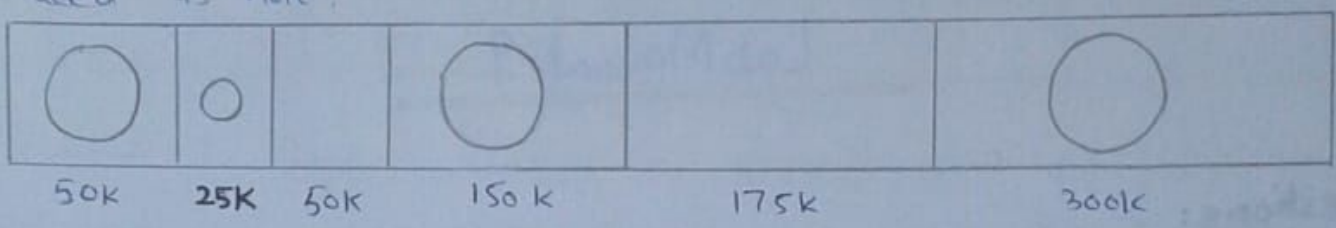
1- First fit algorithm
2- Best fit algorithm
3- Neither of the two
4- Both of them.

In the question, there are five partitions in the memory. 3 partition are having processes inside them and two partitions are holes.
our task is to check the algorithm which can satisfy the request optimally.

Using first fit algorithm:

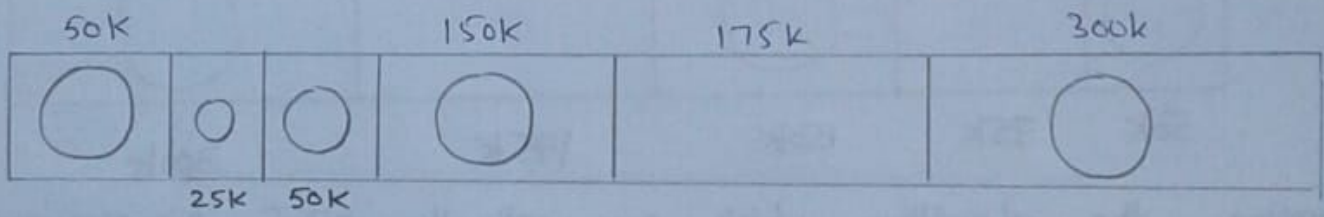
1- 25K requirement:

The algorithm scans the list until it gets first hole which should be big enough to satisfy the request of 25k. It gets the space in the second partition which is free hence it allocates 25k out of 75k to the process and the remaining 50k is produced as hole.



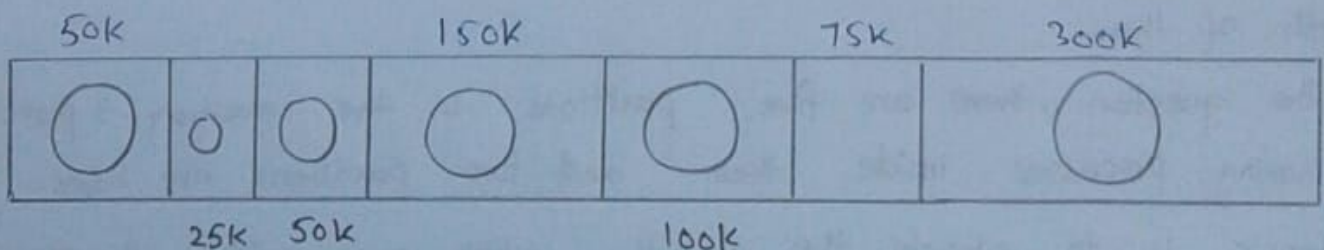
2- 50k requirement:

The 50k requirement can be fulfilled by allocating the third partition which is 50k in size to the process. No free space is produced as free space.



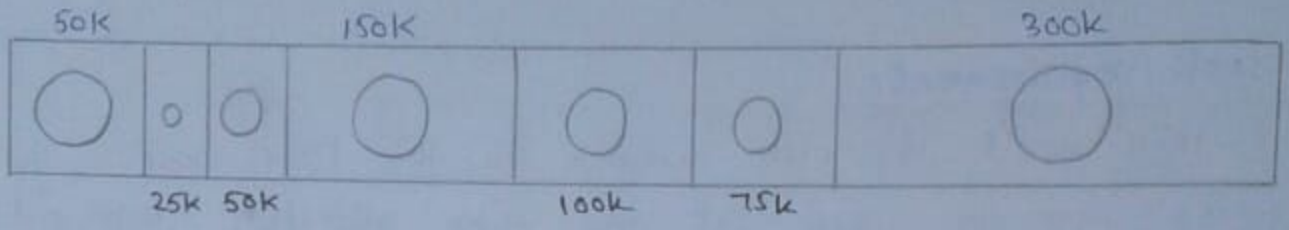
3- 100K requirement:

100k requirement can be fulfilled by using the fifth partition of 175k size, out of 175 K, 100k will be allocated and remaining 75k will there as a hole.



4. 75K requirement:

Since we are having a 75K free partition hence we can allocates that much space to the process which is demanding just 75K space.



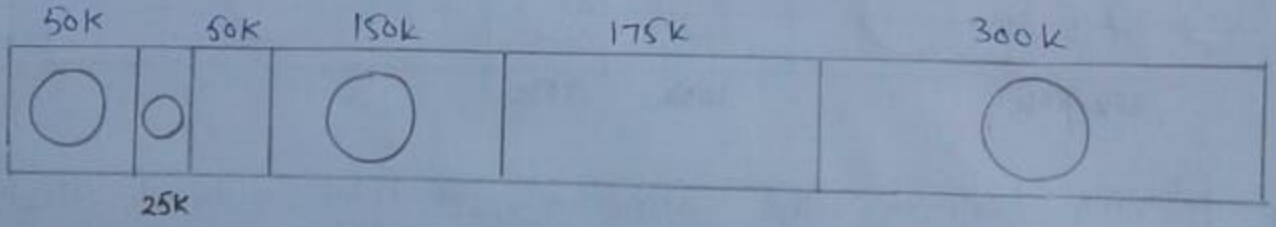
using first fit algorithm, we have fulfilled the entire request optimally and no useless space is remaining.

Using Best fit algorithm:

1. 25K requirement:

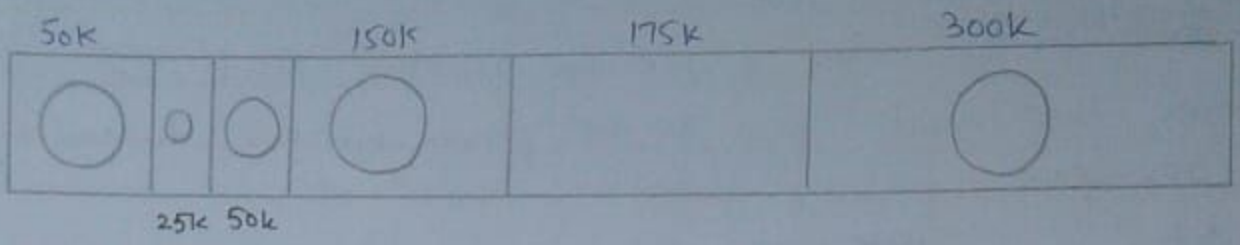
To allocate 25K space using best fit approach, need to scan the whole list and then we find that a 75K partition is free and the smallest among all, which can accomodate the need of the process.

Therefore 25K out of those 75K free partition is allocated to the process and the remaining 50K is produced as a hole.



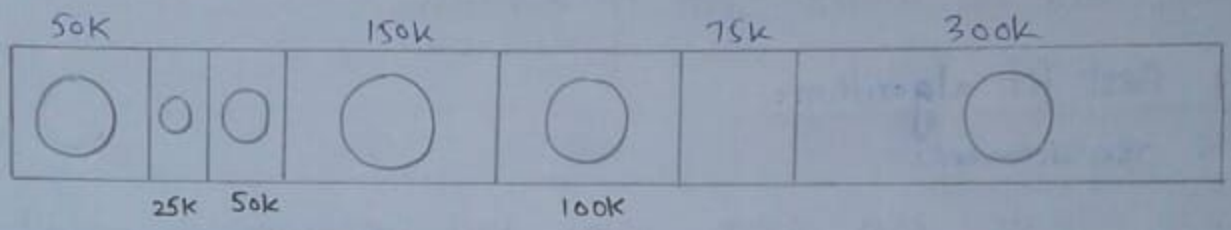
2-50K requirement:

To satisfy this need, we will again scan the whole list and then find the 50K space is free which the exact match of the need is. Therefore, it will be allocated for the process.



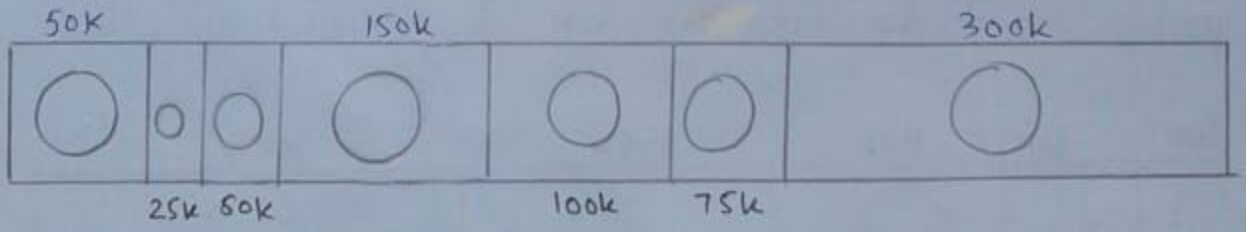
3- 100K requirement:

100K need is close enough to the 175K space. The algorithm scans the whole list and then allocates 100K out of 175K from the 5th free partition.



4. 75K requirement:

75K requirement will get the space of 75K from the 6th free partition but the algorithm will scan the whole list in the process of taking this decision.



By following both of the algorithms, we have noticed that both the algorithms perform similar to most of the extant in the case.

Both can satisfy the need of the processes but however, the best fit algorithm scans the list again and again which takes lot of time

Therefore, if you ask me that which algorithm perfoms in more optimal way then it will be **first Fit algorithm** for sure.

Therefore, the answer in this case is **A.**

Question #2:

Consider five memory partitions of size 100 KB, 500 KB, 200 KB, 450 KB and 600 KB in same order. If sequences of requests for blocks of size 212 KB, 417 KB, 112 KB and 426 KB in same order came, then which of the following algorithm makes the efficient use of memory?

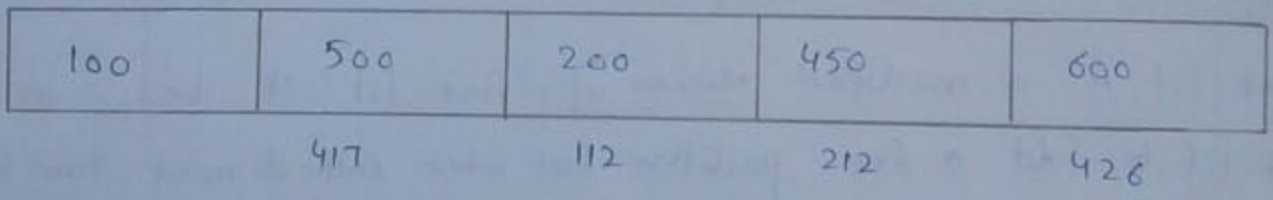
A. Best fit algorithm
B. first fit algorithm
C. Next fit algorithm
D. Both next fit algorithm and best fit algorithm if result are same.

Please provide result.

using Best fit algorithm:

Memory partition: 100KB, 500KB, 200KB, 450KB, 600KB

Requests: 212 KB, 417 KB, 112 KB, 426 KB



The best-fit algorithm selects the partition whose size is closet in size (and large enough) to the requested size.

Memory wastage:

$$(500 - 417) + (200 - 112) + (450 - 212) + (600 - 426)$$

$$= 583 \text{ KB}$$

using first fit algorithm:

Memory partition: 100KB, 500KB, 200KB, 450KB, 600 KB.

Requests: 212, 417, 112, 426.

100	500	200	450	600
	212	112	417	426

The first-fit algorithm selects the first free partition that is large enough to accomodate the request.

Memory wastage =

$(500 - 212) + (200 - 112) + (450 - 417) + (600 - 426) = 583$ kB.

Next fit algorithm:

Memory partition: 100KB, 500KB, 200KB, 450KB, 600 KB

Requests: 212, 417, 112, 426

100	500	200	450	600
	212*		417*	112* .....

I take * as the rover pointer.

Next fit is a modified version of "first fit" It begins as first fit to find a free partition but when called next time it starts searching from where it left off, not from the beginning. This policy makes use of a roving pointer. The pointer moves along the memory chain to search for a next fit. This helps in, to avoid the usage of memory always from the head of the free block chain.

Both next field and best field result are same which gives total 683 space are vacant. so according to me if there are dynamic allocation partition then answer will be 'd' if it is static then answer will be change.

Question #3:

Suppose a fixed partitioning memory system with partitions of 100 500K, 200K, 300K, and 600K (in memory order) exists. All five partitions are currently available. using the best fit algorithm, find how much space exists in this system after processes of 212K, 417K, 112K and 350K (in request order) arrive.

Best fit algorithm:

Memory partition (order): 100K, 500K, 200K, 300K, 600K

Requests order : 212K, 417K, 112K, 350K

100	500	200	300	600
	417	112	212	350

so, total memory space wasted or internal fragmentation

$= (500 - 417) + (200 - 112) + (300 - 212) + (600 - 350)$

$= 83K + 88K + 88K + 250K = 509K$

Question #4:

Assume 140K, 260K, 60K memory is free. What is the total external fragmentation that arises for the following requests 110K, 30K, 210K, 50K using Best-fit policy.

A) 120K

B) 110K

C) 60K

D) 30K

Best fit policy:

Memory partition: 140 K, 260 K, 60K

Requests: 110 K, 30K, 210K, 50K

140	260	60
110	210	30

Total external fragmentation = (140 - 110) + (260 - 210) + (60 - 30)
= 30 + 50 + 30 = 110K.

Question # 5:

Consider five memory partitions of size 100KB, 500 KB, 200KB, 450 KB and 600 KB in same order. If sequence of requests for blocks of size 212KB, 417 KB, 112KB and 426 KB in same order come, then specify the working of how next fit algorithm allocation?

Next fit algorithm working like same as first fit but the difference is while allocating new requests first fit search slot from starting where as Next fit search slot from previous one

100	500	200	450	600

↳ memory

Request : 212, 417, 112, 426

- Allocating request 212 KB at 500KB

100	500	200	450	600
	212			

- Allocating request 417, Next fit starts scan from 200 KB where first fit starts from 100 KB ==> Allocate 450KB slot.

100	500	200	450	600
	212		417	

- Allocating requests 112 kB at 500 kB Next fit starts scan from 600 kB ==> Allocate 600 kB slot

100	500	200	450	600
	212		417	112

- Allocating request 426 kB at 500 KB Next fit starts scan from 100 kB ==> No slot found ==> can't allocate

100	500	200	450	600
	212		417	112