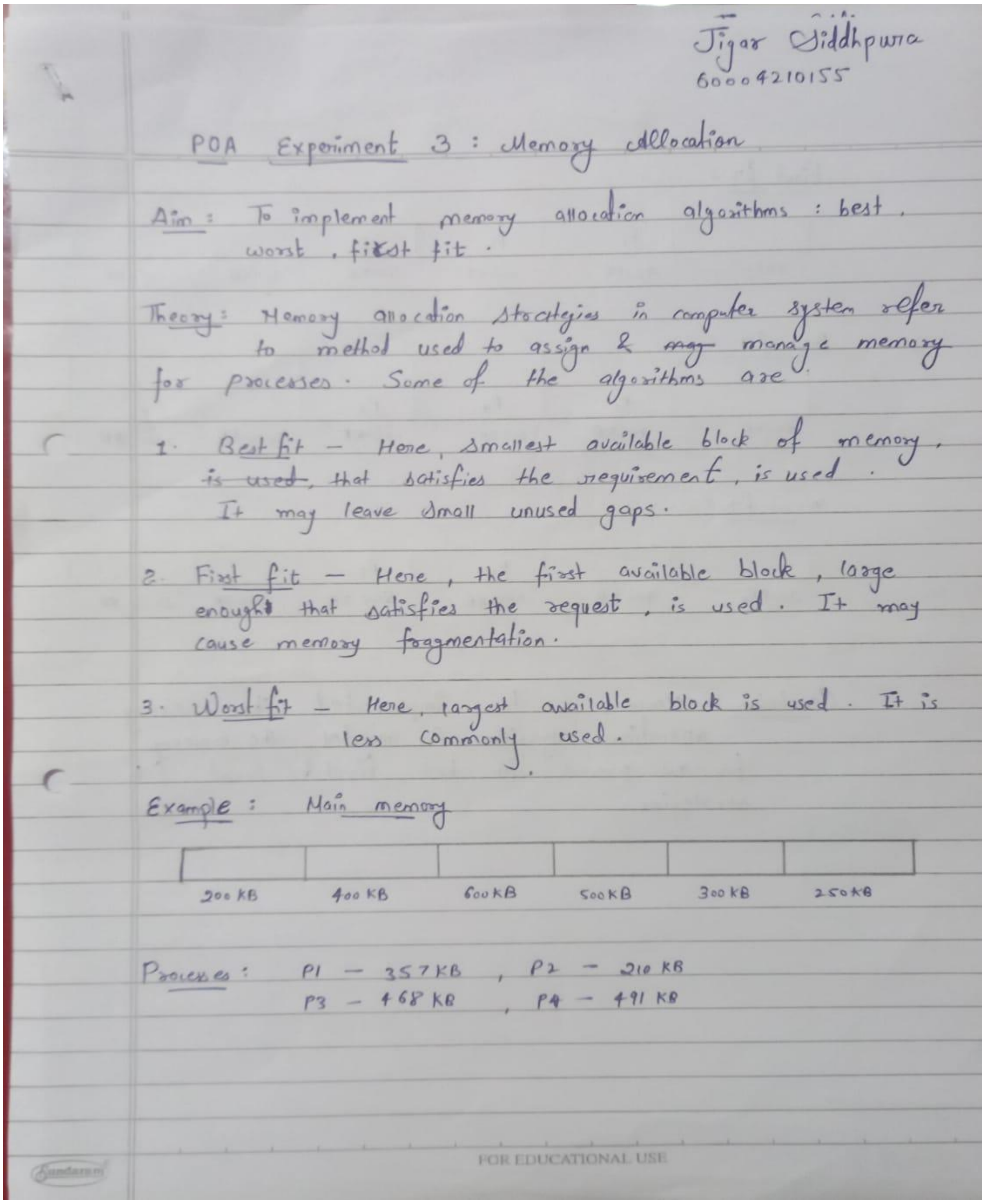
Name: Jigar Siddhpura **SAPID:** 60004200155

DIV: C/C2 **Branch:** Computer Engineering

POA EXPERIMENT 4



Jigar Siddhpura
60004210155

POA Experiment 3 : Memory Allocation

Aim : To implement memory allocation algorithms : best, worst, first fit.

Theory : Memory allocation strategies in computer system refer to method used to assign & ~~mag~~ manage memory for processes. Some of the algorithms are

1. Best fit – Here, smallest available block of memory, ~~is used~~, that satisfies the requirement, is used. It may leave small unused gaps.

2. First fit – Here, the first available block, large enough that satisfies the request, is used. It may cause memory fragmentation.

3. Worst fit – Here, largest available block is used. It is less commonly used.

Example : Main memory

200 KB	400 KB	600 KB	500 KB	300 KB	250 KB

Processes : P1 – 357 KB , P2 – 210 KB
P3 – 468 KB , P4 – 491 KB

First fit :

	P_1	P_2	P_3		
200 KB	400 KB	600 KB	500 KB	300 KB	250 KB

Best fit :

	P_1	P_4	P_3		P_2
200 KB	400 KB	600 KB	500 KB	300 KB	250 KB

Worst fit :

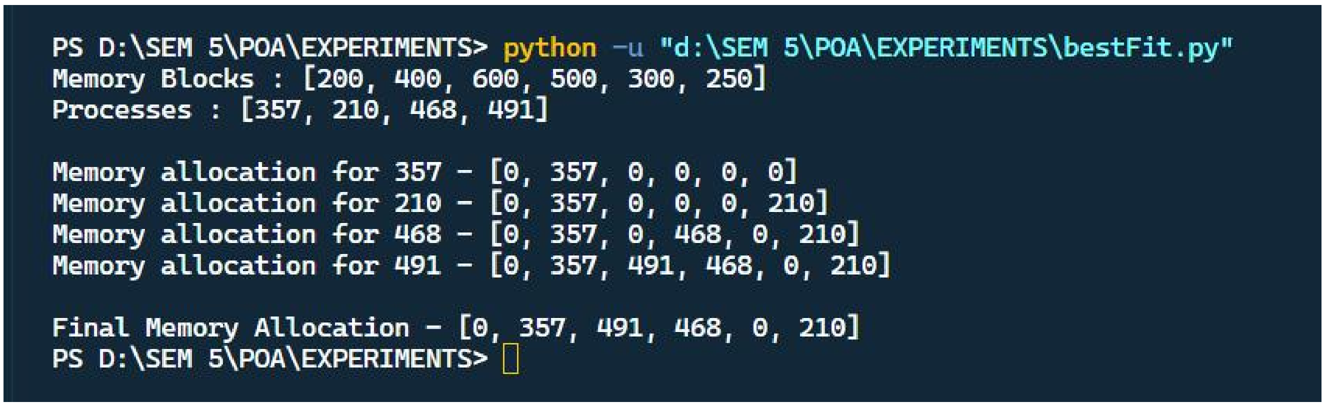
		P_1	P_2		
200 KB	400 KB	600 KB	500 KB	300 KB	250 KB

Conclusion : Hence, we have implemented the memory allocation algorithm & allocated the memory for the process with best, first & worst fit strategies.

Best Fit:

```python
bNo = 6
pNo = 4
bSizes = [200,400,600,500,300,250]
pSizes = [357,210,468,491]
flag = [0] * len(bSizes)
allocated = [0] * len(bSizes)
MAX_BLOCK_SIZE = 999999

print(f"Memory Blocks : {bSizes}")
print(f"Processes : {pSizes}\n")

for i,psize in enumerate(pSizes):
    index_placed = -1
    for j,bsize in enumerate(bSizes):
        if((psize <= bsize) & (bsize < MAX_BLOCK_SIZE) & (flag[j] == 0)):
            index_placed = j
            MAX_BLOCK_SIZE = bsize
    if index_placed != -1:
        flag[index_placed] = 1
        allocated[index_placed] = psize
        print(f"Memory allocation for {psize} - {allocated}")
    else:
        print(f"Memory allocation for {psize} - No Space to allocate")
    MAX_BLOCK_SIZE = 999999

print(f"\nFinal Memory Allocation - {allocated}")
```

Output :

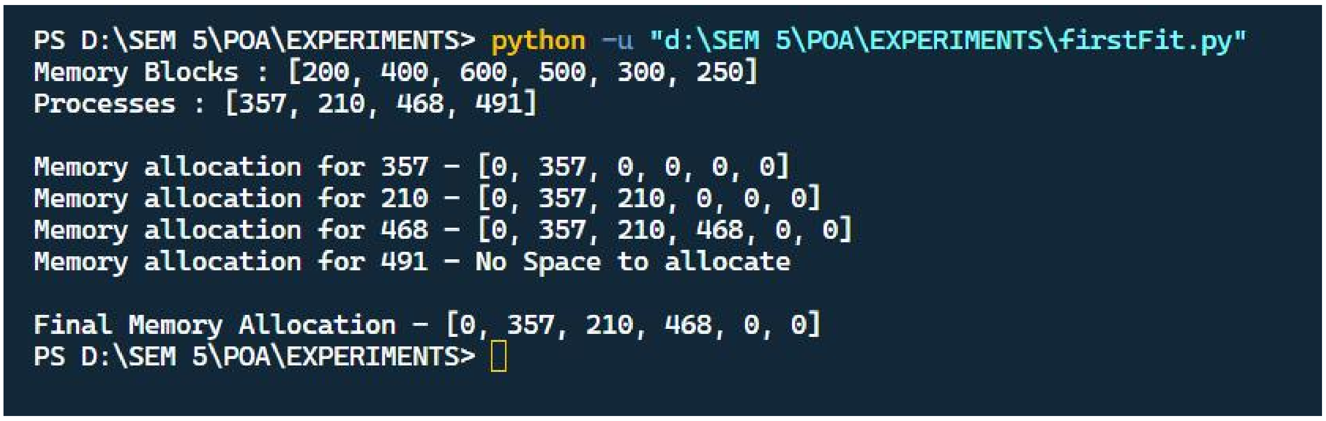


```
PS D:\SEM 5\POA\EXPERIMENTS> python -u "d:\SEM 5\POA\EXPERIMENTS\bestFit.py"
Memory Blocks : [200, 400, 600, 500, 300, 250]
Processes : [357, 210, 468, 491]

Memory allocation for 357 - [0, 357, 0, 0, 0, 0]
Memory allocation for 210 - [0, 357, 0, 0, 0, 210]
Memory allocation for 468 - [0, 357, 0, 468, 0, 210]
Memory allocation for 491 - [0, 357, 491, 468, 0, 210]

Final Memory Allocation - [0, 357, 491, 468, 0, 210]
PS D:\SEM 5\POA\EXPERIMENTS> 
```

First Fit:

```python
bNo = 6
pNo = 4
bSizes = [200,400,600,500,300,250]
pSizes = [357,210,468,491]
flag = [0] * len(bSizes)
allocated = [0] * len(bSizes)

print(f"Memory Blocks : {bSizes}")
print(f"Processes : {pSizes}\n")

for i,psize in enumerate(pSizes):
    for j,bsize in enumerate(bSizes):
        if((psize <= bsize) & (flag[j] == 0)):
            flag[j] = 1
            allocated[j] = psize
            print(f"Memory allocation for {psize} - {allocated}")
            break
        elif(j == len(bSizes)-1):
            print(f"Memory allocation for {psize} - No Space to allocate")

print(f"\nFinal Memory Allocation - {allocated}")
```

Output :

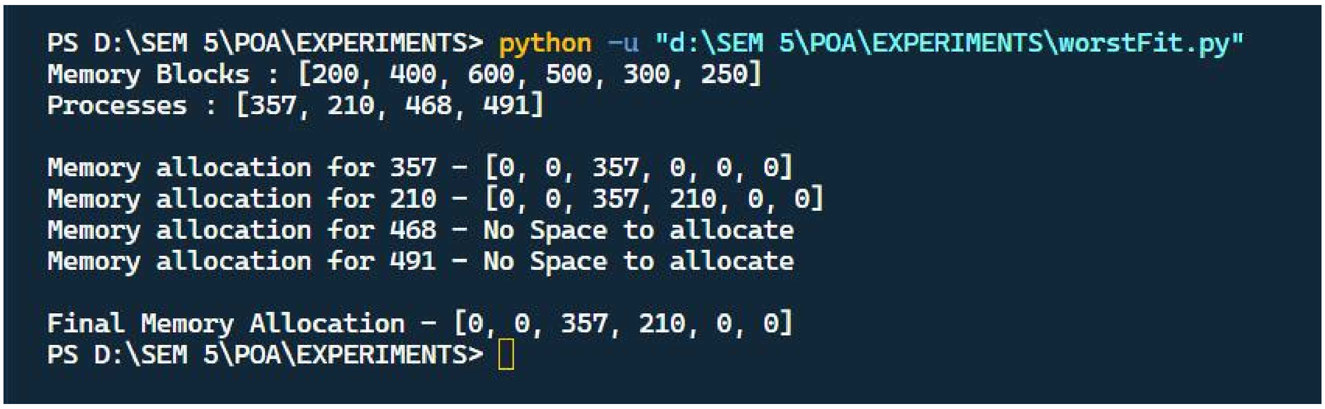


```
PS D:\SEM 5\POA\EXPERIMENTS> python -u "d:\SEM 5\POA\EXPERIMENTS\firstFit.py"
Memory Blocks : [200, 400, 600, 500, 300, 250]
Processes : [357, 210, 468, 491]

Memory allocation for 357 - [0, 357, 0, 0, 0, 0]
Memory allocation for 210 - [0, 357, 210, 0, 0, 0]
Memory allocation for 468 - [0, 357, 210, 468, 0, 0]
Memory allocation for 491 - No Space to allocate

Final Memory Allocation - [0, 357, 210, 468, 0, 0]
PS D:\SEM 5\POA\EXPERIMENTS>
```

Worst Fit :

```python
bNo = 6
pNo = 4
bSizes = [200,400,600,500,300,250]
pSizes = [357,210,468,491]
flag = [0] * len(bSizes)
allocated = [0] * len(bSizes)
MIN_BLOCK_SIZE = -10

print(f"Memory Blocks : {bSizes}")
print(f"Processes : {pSizes}\n")

def getMaxMemoryIndex(psize,memory, flag):
    """
    Returns memory index with max capacity given that it is unallocated.

    Parameters:
    - psize (int): integer representing the process size
    - memory (list): List representing memory capacities.
    - flag (list): List representing the allocation status (0 for unallocated, 1 for allocated).

    Returns:
    - int: Index of the unallocated memory with the maximum capacity but less than process size.
    """
    max_capacity = -1
    max_index = None
    for i in range(len(memory)):
        if flag[i] == 0 and memory[i] >= psize and memory[i]>max_capacity:
            max_capacity = memory[i]
            max_index = I
    return max_index

for i,psize in enumerate(pSizes):
    index_placed = -1
    allocated_index = getMaxMemoryIndex(psize,bSizes,flag)
    if allocated_index:
        flag[allocated_index] = 1
        allocated[allocated_index] = psize
        print(f"Memory allocation for {psize} - {allocated}")
    else:
        print(f"Memory allocation for {psize} - No Space to allocate")

print(f"\nFinal Memory Allocation - {allocated}")
```

Output :



```
PS D:\SEM 5\POA\EXPERIMENTS> python -u "d:\SEM 5\POA\EXPERIMENTS\worstFit.py"
Memory Blocks : [200, 400, 600, 500, 300, 250]
Processes : [357, 210, 468, 491]

Memory allocation for 357 - [0, 0, 357, 0, 0, 0]
Memory allocation for 210 - [0, 0, 357, 210, 0, 0]
Memory allocation for 468 - No Space to allocate
Memory allocation for 491 - No Space to allocate

Final Memory Allocation - [0, 0, 357, 210, 0, 0]
PS D:\SEM 5\POA\EXPERIMENTS> 
```