

Ex. No.: 10a)

Date: 09/04/25

### BEST FIT

Aim:

To implement Best Fit memory allocation technique using Python.

Algorithm:

1. Input memory blocks and processes with sizes
2. Initialize all memory blocks as free.
3. Start by picking each process and find the minimum block size that can be assigned to current process
4. If found then assign it to the current process.
5. If not found then leave that process and keep checking the further processes.

Program Code:

```
def best_fit():
    blocks_input = input("Enter memory block sizes: ")
    blocks = [int(size) for size in blocks_input.split()]

    procs_input = input("Enter process sizes: ")
    procs = [int(size) for size in procs_input.split()]

    alloc = [-1] * len(procs)

    mem = blocks.copy()

    for i in range(len(procs)):
        best = None
        for j in range(len(mem)):
            if mem[j] >= procs[i]:
                if best is None or mem[j] < mem[best]:
                    best = j
        if best is not None:
            alloc[i] = best
```

rem[best] = sizes[i]

print("\n Process No. \t Process Size \t Block No.")

for i in range(len(sizes)):

print(f"{i+1} \t {sizes[i]} \t", end=" ")

if alloc[i] != -1:

print(alloc[i] + 1)

else:

print("Not Allocated")

print("Memory Allocation - Best Fit Algorithm")  
best-fit()

Sample Output:

Process No.	Process Size	Block no.
1	212	4
2	417	2
3	112	3
4	426	5

Memory Allocation - Best Fit Algorithm  
Enter memory block size : 100 200 300  
Enter process sizes : 150 50 250

Process no.	Process size	Block no.
1	150	2
2	50	2
3	250	3

Result:

Thus the python program to implement Best Fit memory allocation technique has been executed successfully.

