

**Memory Compaction** Memory compaction is time-consuming, and the operating system must decide efficiently how to assign processes to memory blocks.

### Placement Algorithms:

1. **Best-Fit:** Chooses the block closest in size to the request.
2. **First-Fit:** Scans from the beginning and selects the first available block large enough.
3. **Next-Fit:** Starts scanning from the last allocated location and selects the next suitable block.
4. **Worst-Fit:** Chooses the largest empty block available.

### Comparison of Algorithms:

- **First-Fit:** Fastest and simplest; may cause small free partitions at the front.
  - **Next-Fit:** Similar to First-Fit but results in breaking large blocks at the end more often.
  - **Best-Fit:** Often the worst performer as it creates many small unusable fragments.
  - **Worst-Fit:** Leaves large blocks available but is not always efficient.
- 

### First-Fit Algorithm:

1. Start
  2. Accept the number of partitions (n) and their sizes.
  3. Accept the number of processes (p) and their sizes.
  4. Initialize flag array to 0.
  5. For each partition, check if a process fits and assign it.
  6. Mark the process as allocated and print allocation.
  7. End
- 

### Best-Fit Algorithm:

1. Start
2. Accept the number of partitions (n) and their sizes.
3. Accept the number of processes (p) and their sizes.
4. Sort the partitions in ascending order.

5. Initialize flag array to 0.
  6. For each partition, check if a process fits and assign it.
  7. Mark the process as allocated and print allocation.
  8. End
- 

## Worst-Fit Algorithm:

1. Start
2. Accept the number of partitions (n) and their sizes.
3. Accept the number of processes (p) and their sizes.
4. Sort the partitions in descending order.
5. Initialize flag array to 0.
6. For each partition, check if a process fits and assign it.
7. Mark the process as allocated and print allocation.
8. End

## Conclusion:

- **First-Fit** is usually the best choice for speed and efficiency.
- **Best-Fit** leads to memory fragmentation issues.
- **Worst-Fit** may keep large blocks free but isn't always efficient.
- **Next-Fit** tends to waste larger blocks at the end of memory space.