| **First Fit** | **Best Fit** | **Worst Fit** | **Fragmentation** | |
| --- | --- | --- | --- | --- |
| **External** | **Internal** |
| Allocate the first free partition or hole large enough which can accommodate the process. | Allocating the smallest free partition which meets the requirement of the requesting process | The process traverses the whole memory and always searches for the largest hole, and then the process is placed in that hole. | Dynamic memory allocates memory but leaves a small amount of memory unusable | If the process is smaller than the amount of memory requested, a free space is created in the given memory block. |
| **Advantages:**   * Fastest memory allocation to the coming processes as it allocates the largest first fit algorithm to the new processes. * Helps to minimize memory fragmentation, as it tends to allocate memory in larger blocks. * Simplicity and efficiency | **Advantages:**   * It takes less time compared to the other algorithms. * Improved memory utilization. * It is easiest to implement. * Reduce the wastage of space because it allocated the smallest appropriate holes for requesting process | **Advantages:**   * Reduce the problem of fragmentation by allocating the largest fragments to new processes. | **Advantages:**   * Data write in a system may be faster than reorganizing data storage to enable contiguous data writes. * Fewer Failures as If there is insufficient sequential space in a system that does not support fragmentation, the write will fail. * Make better use of a storage device by utilizing every available storage block. | |
| **Disadvantages:**   * Poor performance in situations where the memory is highly fragmented. * The search for a suitable block of memory can become time-consuming and inefficient. * Poor memory utilization | **Disadvantages:**   * Increased computational overhead (it requires that the entire list be searched). * Increased fragmentation, as it may leave smaller blocks of memory scattered throughout the memory space. | **Disadvantages:**   * It is a slow process because it traverses all the partitions in the memory and then selects the largest partition among all the partitions, which is a time-consuming process. | **Disadvantages:**   * Need for regular defragmentation as more fragmented storage device's performance will degrade with time. * Slower Read Times as The time it takes to read a non-sequential file might increase as a storage device becomes more fragmented. | |