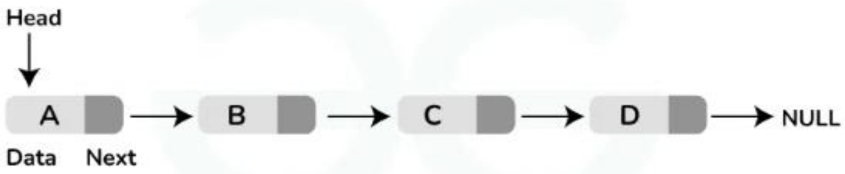
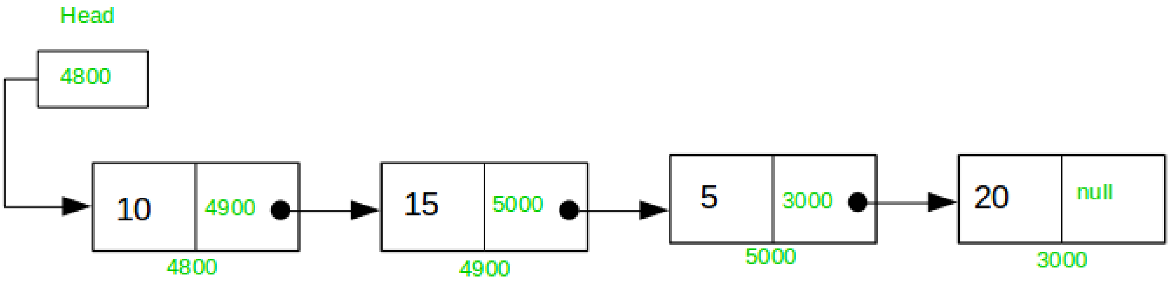
Singly Linked List

What is Singly Linked List?

*A **singly linked list** is a linear data structure in which the elements are not stored in contiguous memory locations and each element is connected only to its next element using a pointer.*



Singly Linked list



Characteristics of a Singly Linked List:

- Each node holds a single value and a reference to the next node in the list.
- The list has a head, which is a reference to the first node in the list, and a tail, which is a reference to the last node in the list.
- The nodes are not stored in a contiguous block of memory, but instead, each node holds the address of the next node in the list.
- Accessing elements in a singly linked list requires traversing the list from the head to the desired node, as there is no direct access to a specific node in memory.

Application of Singly Linked Lists:

- **Memory management:** Singly linked lists can be used to implement memory pools, in which memory is allocated and deallocated as needed.
- **Database indexing**: Singly linked lists can be used to implement linked lists in databases, allowing for fast insertion and deletion operations.
- **Representing polynomials and sparse matrices:** Singly linked lists can be used to efficiently represent polynomials and sparse matrices, where most elements are zero.
- **Operating systems:** Singly linked lists are used in operating systems for tasks such as scheduling processes and managing system resources.

Advantages of Singly Linked Lists:

- **Dynamic memory allocation**: Singly linked lists allow for dynamic memory allocation, meaning that the size of the list can change at runtime as elements are added or removed.
- **Cache friendliness:** Singly linked lists can be cache-friendly as nodes can be stored in separate cache lines, reducing cache misses and improving performance.
- **Space-efficient:** Singly linked lists are space-efficient, as they only need to store a reference to the next node in each element, rather than a large block of contiguous memory.

Disadvantages of Singly Linked Lists:

- **Poor random access performance**: Accessing an element in a singly linked list requires traversing the list from the head to the desired node, making it slow for random access operations compared to arrays.
- **Increased memory overhead:** Singly linked lists require additional memory for storing the pointers to the next node in each element, resulting in increased memory overhead compared to arrays.
- **Vulnerability to data loss:** Singly linked lists are vulnerable to data loss if a node’s next pointer is lost or corrupted, as there is no way to traverse the list and access other elements.
- **Not suitable for parallel processing:** Singly linked lists are not suitable for parallel processing, as updating a node requires exclusive access to its next pointer, which cannot be easily done in a parallel environment.
- **Backward traversing not possible:** In singly linked list does not support backward traversing.