**Linked List Cheat Sheet**

**Inserting data to a Linked List**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **SLL** | **CSLL** | **DLL** | **CDLL** |
| **LL is empty** | head = node  tail = node | head = node  tail = node  node.next = node | head = node  tail = node | head = node  tail = node  node.prev = node  node.next = node |
| **Position = 1** | node.next = head  head = node | node.next = head  head = node  tail.next = node | head.prev = node  node.next = head  head = node | head.prev = node  tail.next = node  node.next = head  node.prev = tail  head = node |
| **Position = -1** | tail.next = node  tail = node | tail.next = node  tail = node  node.next = head | tail.next = node  node.prev = tail  tail = node | tail.next = node  head.prev = node  node.prev = tail  node.next = head  tail = node |
| **Position = n** | prv\_nd = head  cur\_nd = head.next  for i in range(1, pos-1)  prv\_nd = prv\_nd.next  cur\_nd = cur\_nd.next  node.next = cur\_nd  prv\_nd.next = node | | prv\_nd = head  cur\_nd = head.next  for i in range(1, pos-1)  prv\_nd = prv\_nd.next  cur\_nd = cur\_nd.next  node.prev = prv\_nd  node.next = cur\_nd  prv\_nd.next = node  cur\_nd.prev = node | |
| **length += 1** | | | | |

**Deleting data from a Linked List**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **SLL** | **CSLL** | **DLL** | **CDLL** |
| **Empty the LL** | head = None  tail = None | tail.next = None  head = None  tail = None | cur\_nd = head  while cur\_nd:  cur\_nd.prev = None  cur\_nd = cur\_nd.next  head = None  tail = None | tail.next = None  cur\_nd = head  while cur\_nd:  cur\_nd.prev = None  cur\_nd = cur\_nd.next  head = None  tail = None |
| **length = 0** | | | | |
|  | **SLL** | **CSLL** | **DLL** | **CDLL** |
| **Position = 1 or -1**  **head = tail** | head = None  tail = None | tail.next = None  head = None  tail = None | head = None  tail = None | head.prev = None  tail.next = None  head = None  tail = None |
| **Position = 1** | head = head.next | head = head.next  tail.next = head | head = head.next  head.prev = None | head = head.next  head.prev = tail  tail.next = head |
| **Position = -1** | pos\_ub = length if pos == -1 else pos  prv\_nd = head  cur\_nd = head.next  for i in range(1, pos\_ub-1)  prv\_nd = prv\_nd.next  cur\_nd = cur\_nd.next  if cur\_nd == tail:  tail = prv\_nd  prv\_nd.next = cur\_nd.next | | tail = tail.prev  tail.next = None | tail = tail.prev  tail.next = head  head.prev = tail |
| **Position = n** | prv\_nd = head  cur\_nd = head.next  for i in range(1, pos-1)  prv\_nd = prv\_nd.next  cur\_nd = cur\_nd.next  prv\_nd.next = cur\_nd.next  if cur\_nd == tail:  tail = prv\_nd  else:  cur\_nd.next.prev = prv\_nd | prv\_nd = head  cur\_nd = head.next  for i in range(1, pos-1)  prv\_nd = prv\_nd.next  cur\_nd = cur\_nd.next  if cur\_nd == tail:  tail = prv\_nd  prv\_nd.next = cur\_nd.next  cur\_nd.next.prev = prv\_nd |
| **length -= 1** | | | | |