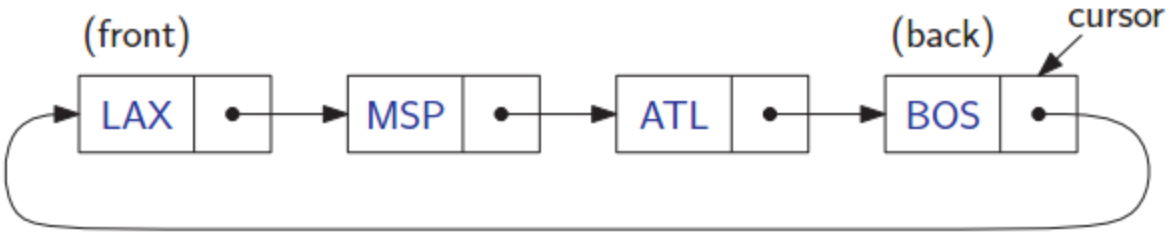
Circularly Linked Lists and List Reversal

Jyh-Shing Roger Jang (張智星)
CSIE Dept, National Taiwan University

Circularly Linked Lists

- A circularly linked list (CLL) allows us to traverse the list in a circular manner
- A node has 2 member variables
  - element
  - link to the next node
- A cursor to indicate where to start the traversal.
  - If we cut the link between the node referenced by the cursor and this node's immediate successor, the result would be a singly linked list from the front to the back node.



More about CLL

- Methods for CLL
  - front()
  - back()
  - advance()
  - add(e)
  - Remove()
- Example
  - Playlist maintenance

Reverse a DLL

- Reverse a DLL

```
void listReverse(DLinkedList& L) {          // reverse a list
  DLinkedList T;                            // temporary list
  while (!L.empty()) {                      // reverse L into T
    string s = L.front();  L.removeFront();
    T.addFront(s);
  }
  while (!T.empty()) {                      // copy T back to L
    string s = T.front();  T.removeFront();
    L.addBack(s);
  }
}
```

Code Fragment 3.35: A function that reverses the contents of a doubly linked list L.

- How to
  - Reverse SLL & CLL
  - Do it in-place