

Linked Lists

Doubly-Linked Lists

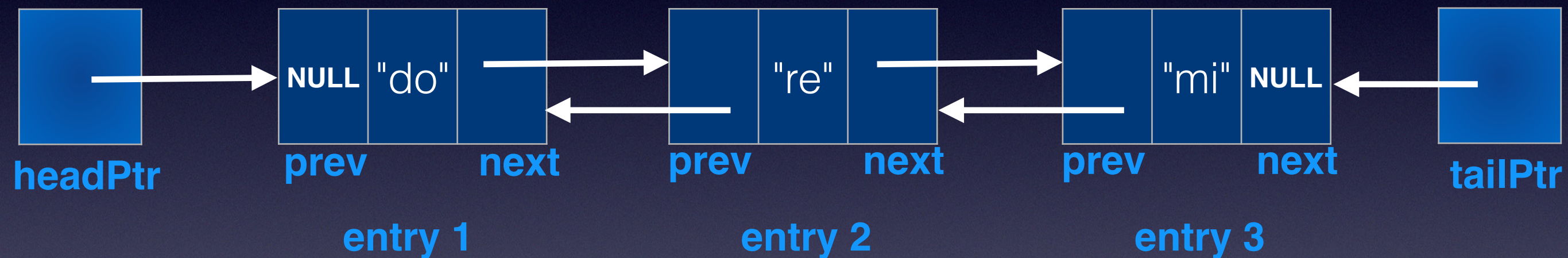
CS110C

Max Luttrell, CCSF

showdown: linked lists vs. array-based lists

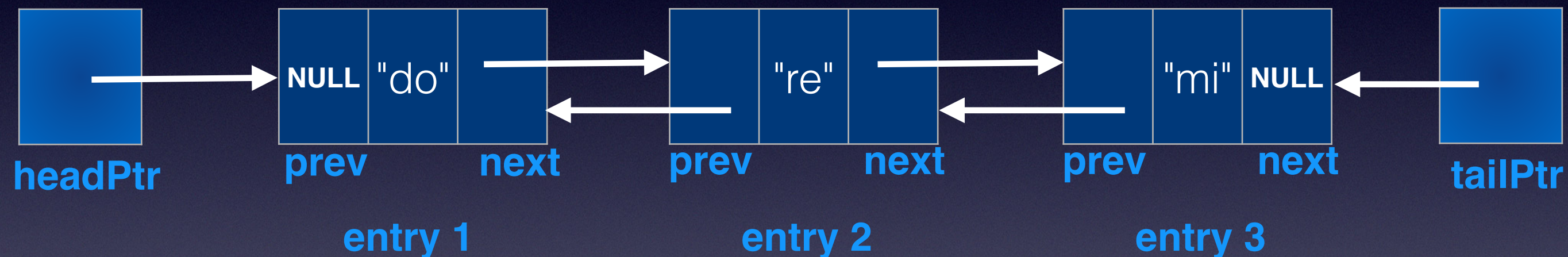
operation	array-based list	linked list	advantage
insertion / deletion	slow - can require shifting many items	fast	linked list
insertion into full list	very slow - requires resizing of array	fast	linked list
random access	fast	slow - requires traversal of list	array-based list
sequential access	fast, can benefit from cache	can be slower (limited cache benefit)	array-based list
storage efficiency	only stores the data	pointer overhead plus data	array-based list

doubly-linked list



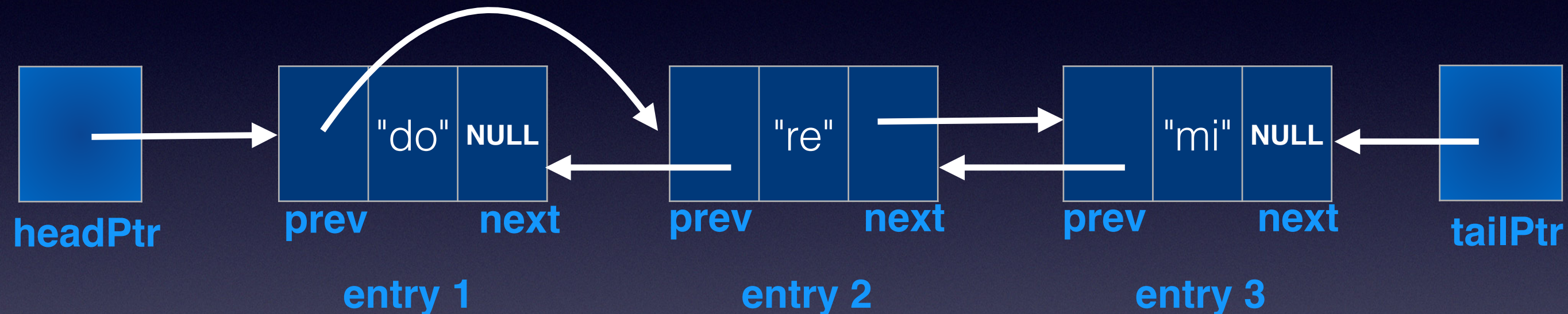
Reverse a doubly-linked list

- First, swap each node's prev/next pointers



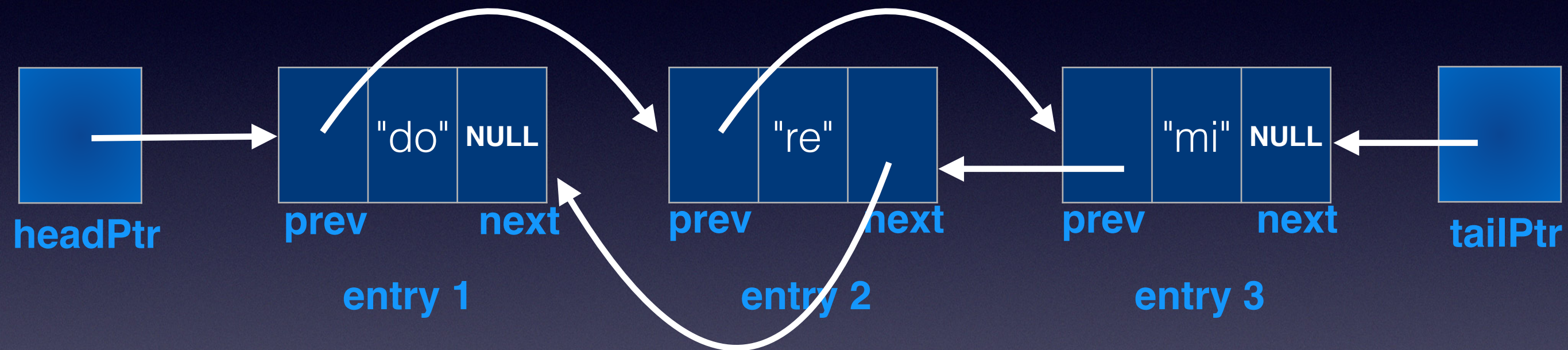
Reverse a doubly-linked list

- First, swap each node's prev/next pointers



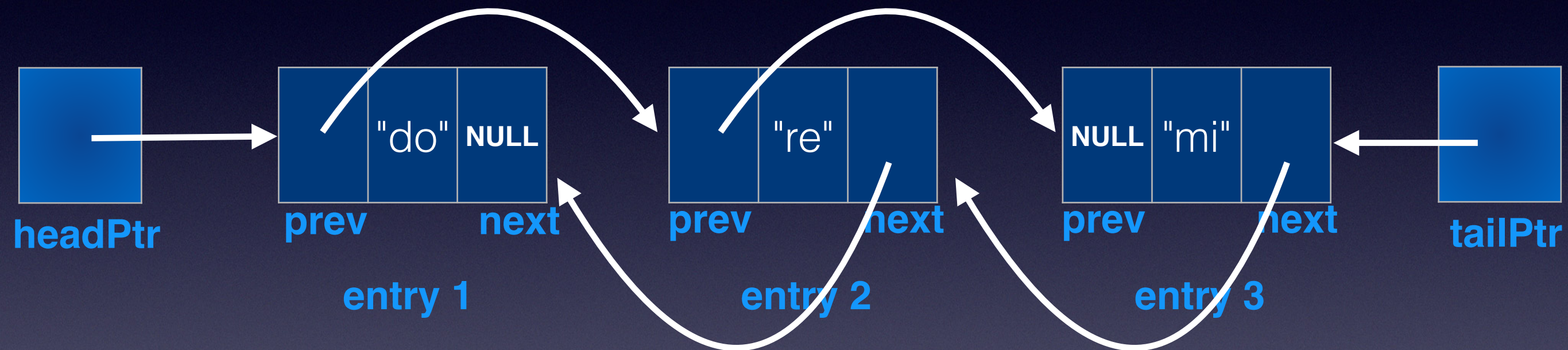
Reverse a doubly-linked list

- First, swap each node's prev/next pointers



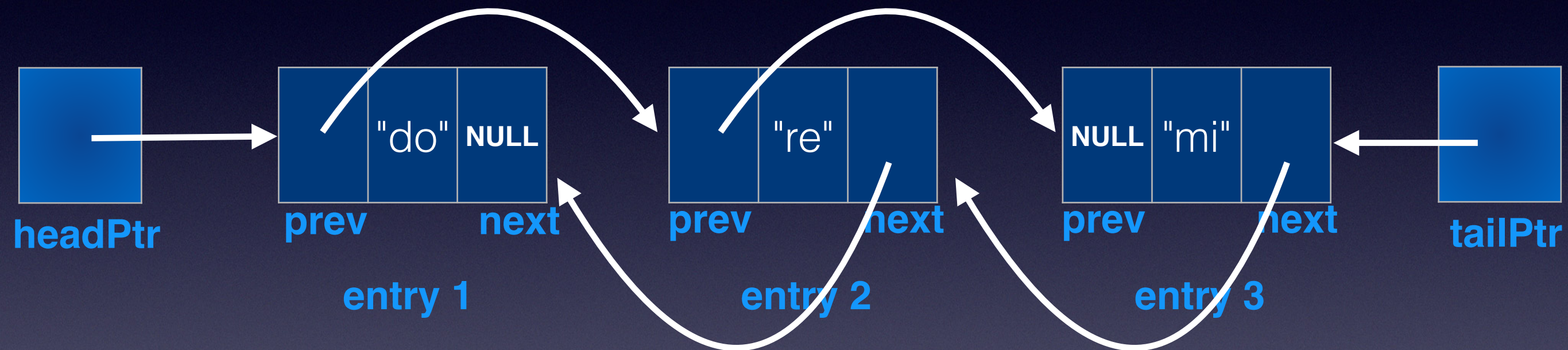
Reverse a doubly-linked list

- First, swap each node's prev/next pointers



Reverse a doubly-linked list

- First, swap each node's prev/next pointers
- Then, swap headPtr and tailPtr



Reverse a doubly-linked list

- First, swap each node's prev/next pointers
- Then, swap headPtr and tailPtr

