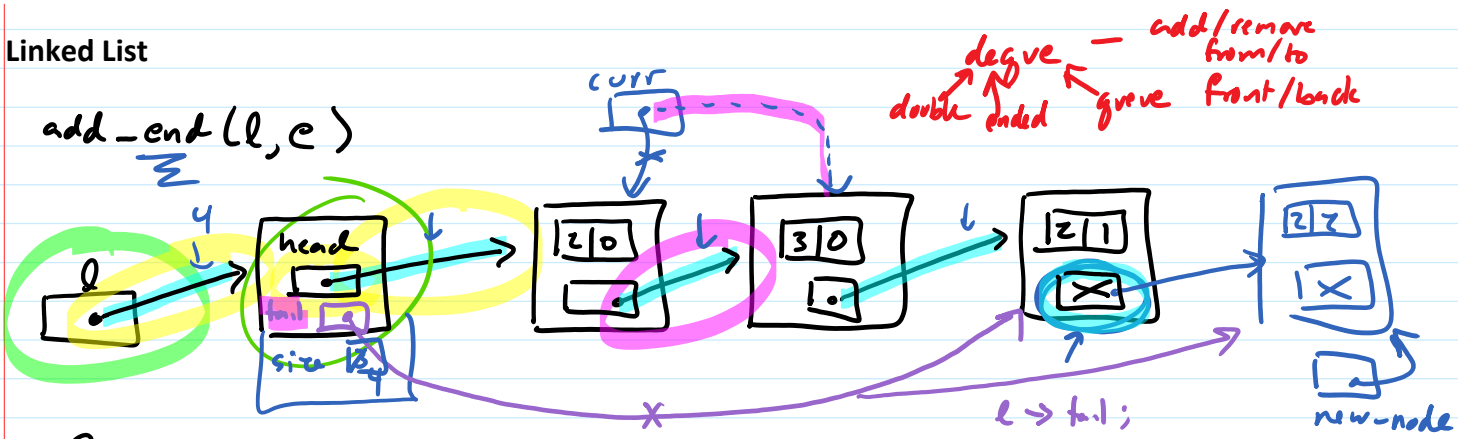
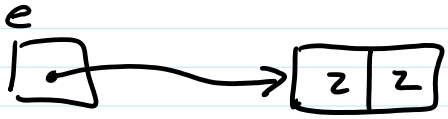
Linked List





```
while ( list_size(l) > 0)
{
    // O(1) inside
}
```

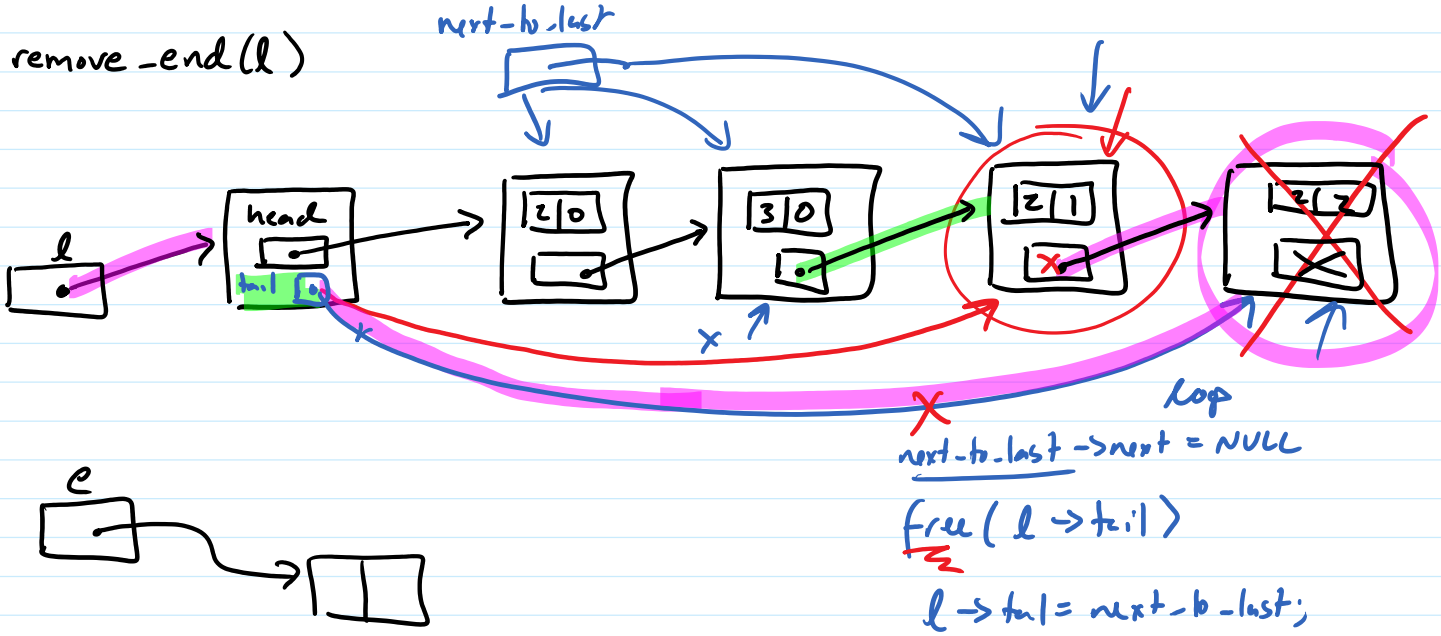
O(n)

add to end of array list
worst case O(n)
amortized O(1)

```
node * curr = l->head;
while ( curr->next != NULL)
    curr = curr->next;
curr->next = new_node
l->tail    = new_node
```

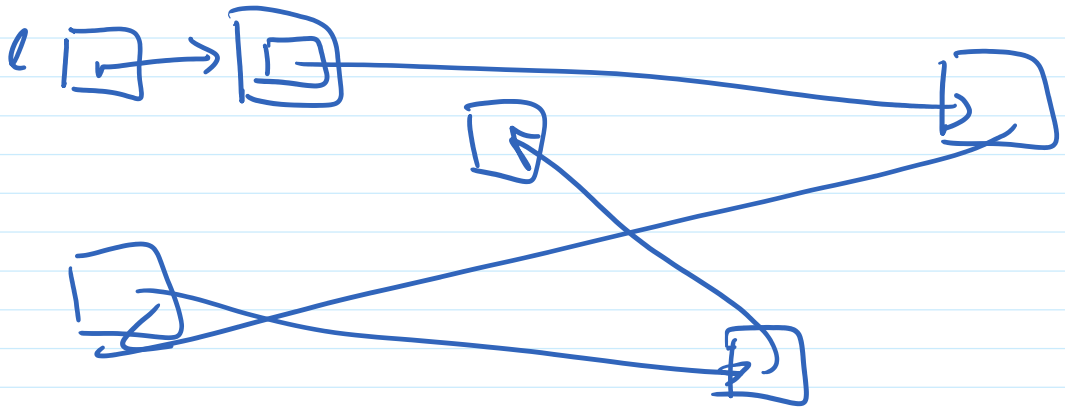
O(1) O(n)

add to end of linked list
worst case O(1)

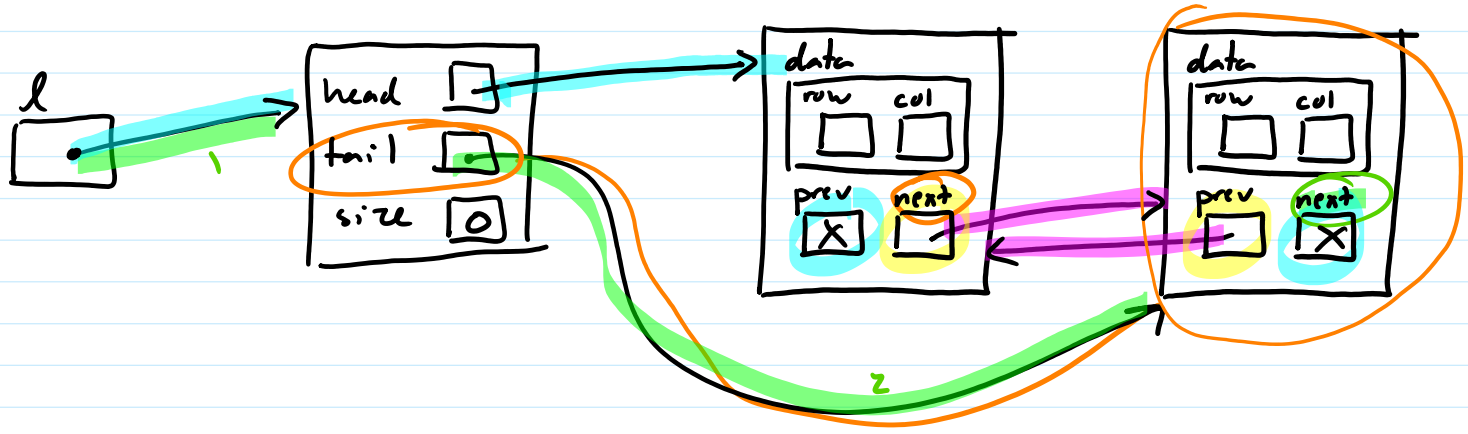



```
next_to_last->next = NULL
free(l->tail)
l->tail = next_to_last;
```

(l->tail) - 1



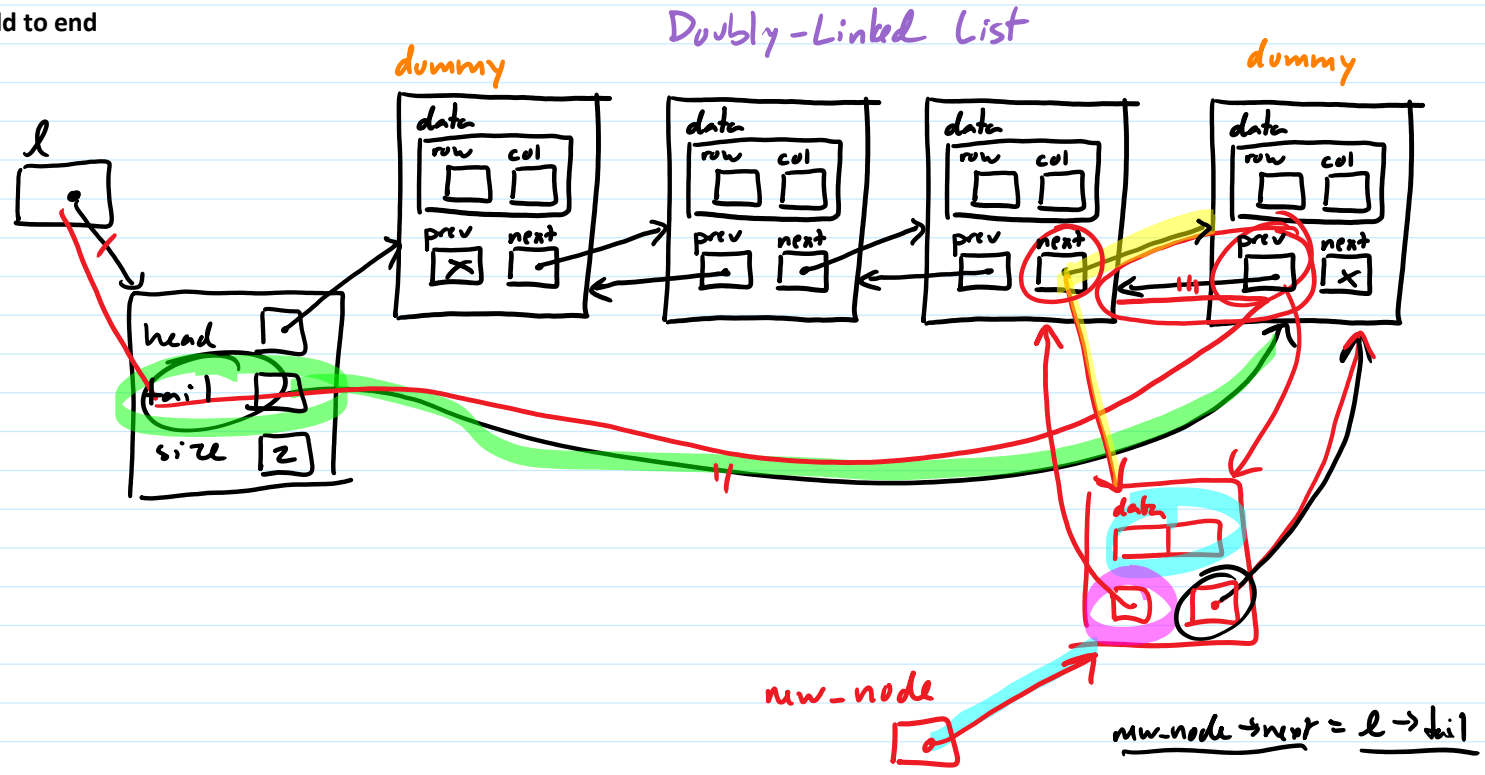
Create



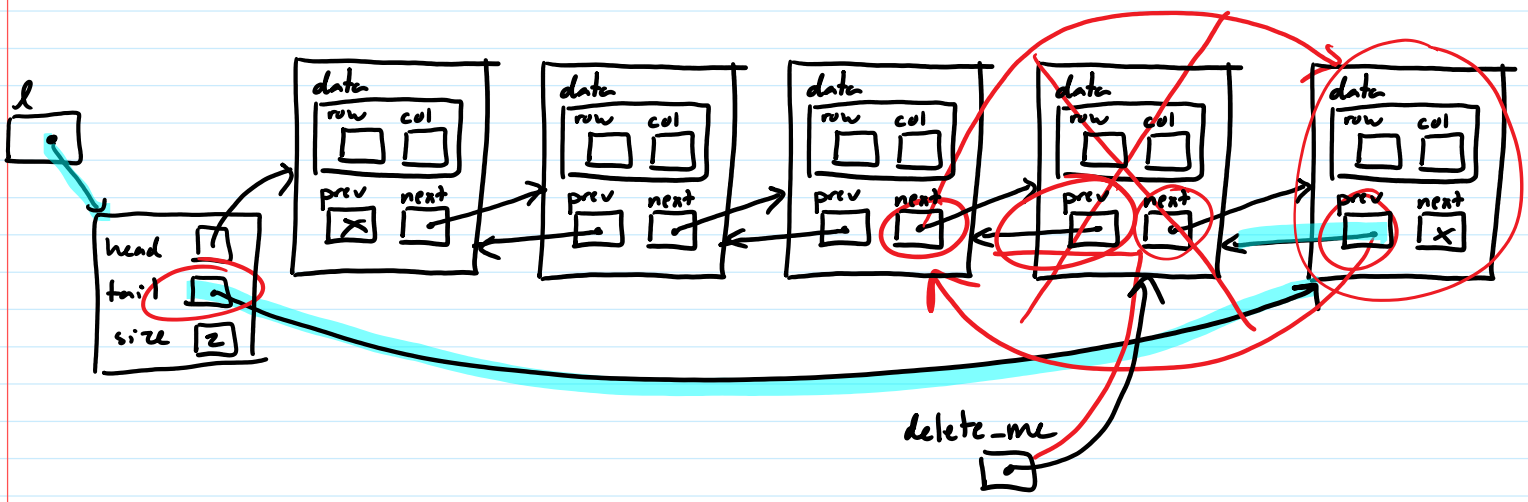
l → head → prev = NULL
l → tail → next = NULL
1 2

l → head → next = l → tail

Add to end


Doubly-Linked List
dummy
dummy
l
data
row
col
prev
next
head
tail
size
2
new_node
new_node->next = l->tail

Remove from end



Get