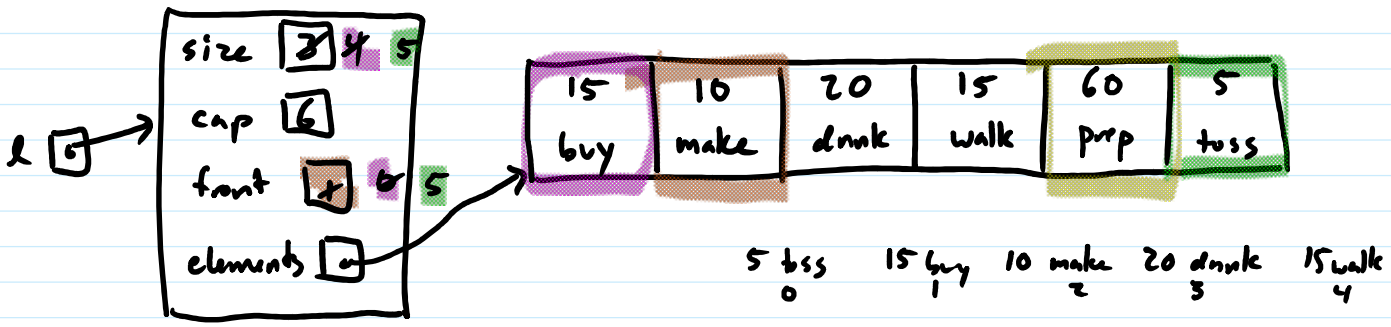
EXAM #1 SCL 110 Wed Oct 11 11:35am-12:50pm

	array list	doubly-linked list	wraparound array
add to back	O(1) if no resize	O(1)	O(1) if no resize
add to front	O(n)	O(1)	O(1)
remove from back	O(1)	O(1)	O(1)
remove from front	O(n)	O(1)	O(1)
add/remove at index	O(n)	O(n)	O(n)
get	O(1)	O(n)	O(1)
size	O(1)	O(1)	O(1)
sort	O(n log n) heapsort or modified quicksort	O(n log n) mergesort	O(n log n)

queue: list restricted to adding at one end, removing from other end
(enqueue, dequeue)

stack: list restricted to adding at one end, removing from same end
(push, pop)

wraparound array

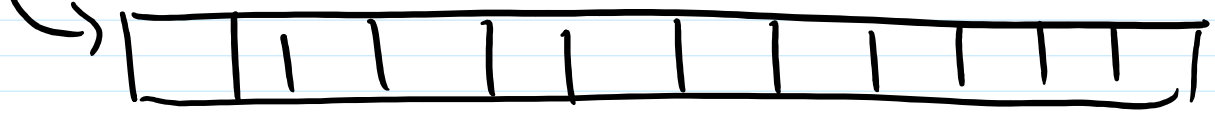


msgs

```
task_list_add_front(l, 15, "buy milk");
task_list_add_front(l, 5, "toss old milk");
task_list_add_back(l, 60, "prep 223 notes");
```

```
task_list_add_back(l, 60, "prep 223 notes");
task_list_get(l, 3, &t);
```

(5 + 3) % 6 = 2



Aggregate analysis (a form of amortized analysis):

compute worst-case for a sequence of operations and average time per operation in the sequence

array list, initial cap 3, double size when needed

1st add
2nd add
3rd add
4th add — 3+1 = 4 resize add — copy 3
5th add
6th add
7th add — 6+1 = 7 — copy 6
8th add
⋮
12th add
13th add — 12+1 = 13 — copy 12
⋮
$n = 3 \cdot 2^k + 1$ nth add — $3 \cdot 2^k + 1 = n$ — copy $3 \cdot 2^k$

total copies over sequence of n adds

$$n + 3 + 6 + 12 + \cdots + 3 \cdot 2^k$$
$$= n + 3(1 + 2 + \cdots + 2^k)$$
$$= n + 3(2^{k+1} - 1)$$
$$= n + 3 \cdot 2^{k+1} - 3$$
$$= n + 2 \cdot 3 \cdot 2^k + 2 - 2 - 3$$
$$= n + 2(3 \cdot 2^k + 1) - 5$$
$$= n + 2n - 5$$
$$= 3n - 5$$

$O(n)$ copies total

avg copies per add $= \frac{3n-5}{n}$
$= 3 - \frac{5}{n}$
< 3
$O(1)$

array list, initial cap 30, add 30 when needed

1st – 30th adds — 1 copy each
31st add — 30+1 = 31 — copy 30
32nd – 60th add — 1 each
61st add — 60+1 = 61 — copy 60
62nd – 90th add — 1 each
⋮
$n = 30 \cdot l + 1$ nth add — $30l + 1 = n$ — copy $30l$

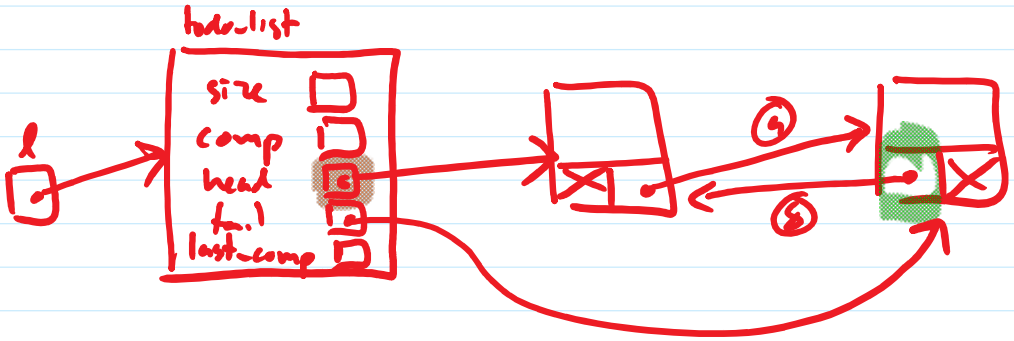
$l = \frac{n-1}{30}$

$$n + (30 + 60 + \cdots + 30l)$$
$$= n + 30 \cdot (1 + 2 + \cdots + l)$$
$$= n + 30 \cdot \frac{l(l+1)}{2}$$
$$= n + 30 \cdot \frac{\left(\frac{n-1}{30}\right)\left(\frac{n-1}{30} + 1\right)}{2}$$

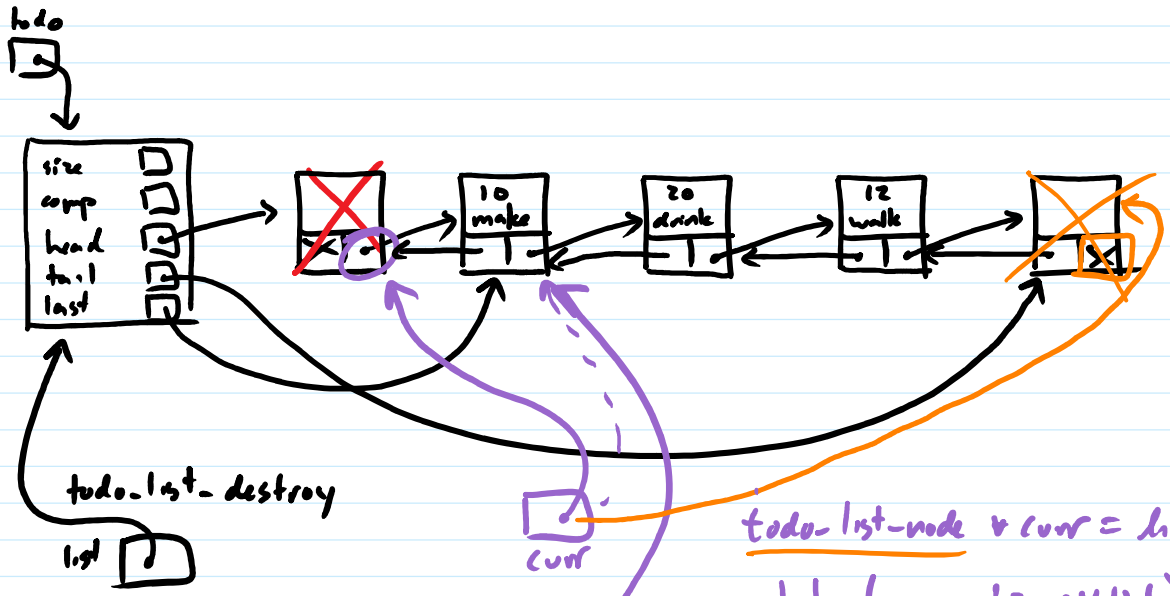
$O(n^2)$ total over sequence of n adds

avg $O(n)$ per add

todo_list_create()




```
4  l->tail->prev = l->head;
5  l->head->next = l->tail;
   l->tail->next = NULL;
   l->head->prev = NULL;
```




```
todo_list_node * curr = list->head;
while ( curr != NULL)
{
    todo_list_node *temp = curr->next;
    free(curr)
    curr = temp;
}
free( list);
```