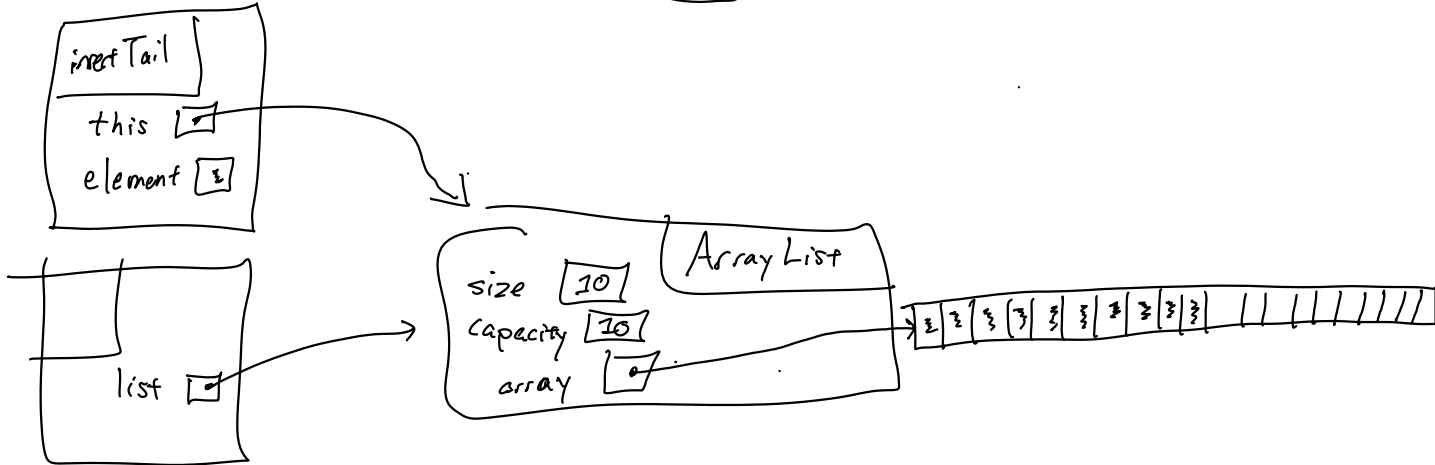


List ADT ← abstract data type

	Linked List w/ size invariant	arrays	Array List
get Size ()	$O(1)$	n/a	
insert First (T element)	$O(1)$	n/a	
remove First ()	$O(1)$	n/a	
get (int i)	$O(n)$	$O(1)$	
insert Last (T element)	$O(n)$	n/a	$O(n)$



Assume initial capacity is 2

	Size	Cap	Steps	Payment	Credit
insert →	0	2	1	1+2	0
insert →	1	2	1	1+2	2
insert →	2	2	1	1+2	4
* insert →	3	4	2+1	2+1-2+2	4
insert →	4	4	1	1+2	6
* insert →	5	8	4+1	4+1-4+2	4
insert →	6	8	1	1+2	6
→	7	8	1	1+2	8
→	8	8	1	1+2	10
→	9	16	8+1	8+1-8+2	2

worst case $O(n)$
amortized worst case $O(1)$

mean cost of all inserts
 $O(1)$ each



Store everywhere we can go (in a list) .

Remember where we've been