

2. insert/delete

insert last $O(1)$ retrieve $O(1)$
 delete last $O(1)$

insert $O(h)$ delete $O(h)$

can't delete
 insert first $O(1)$
 insert last $O(h)$

delete first $O(h)$
 delete last $O(h)$

insert/delete first $O(h)$
 insert last $O(1)$
 delete last $O(h)$

insert/delete first	$O(h)$	$O(1)$
insert last	$O(1)$	$O(h)$
delete last	$O(h)$	$O(h)$
insert/delete (i)	$O(\min\{i+1, h-i+1\})$	$O(i+1)$
retrieve (i)	$O(1)$	$O(i+1)$

Concept of linked list

$$O(n)$$

Concept

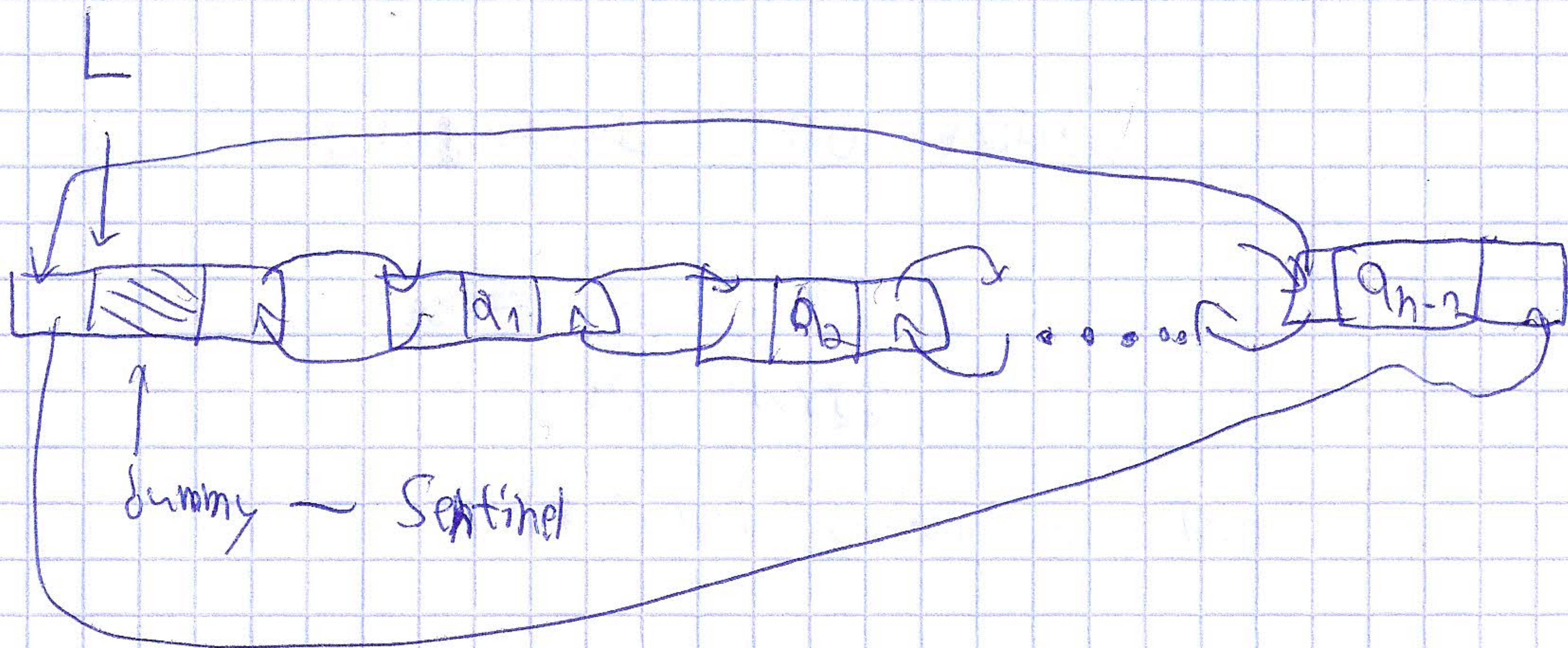
$$O(\min\{h_1, h_2\} + 1)$$

$$O(n)$$

Linked list structure

~~Linked list~~

Doubly linked list



Linked list structure

Linked list structure

2	6	7	5	Null	Null	9	8	4	3
Shalom		Shela		Moshe			Yotam	Dit	
0	1	2	3	4	5	6	7	8	9

List 0
free 1

with resizing
 array as we grow

for resizing we need to copy all elements to a new array of size $2 \times \text{current size}$

if we do this n times, the total cost is $O(n^2)$

if we use a dynamic array, the cost of resizing is amortized to $O(1)$

average case: $O(1)$

$$1 + 1 + 1 + 1 + \dots + 1$$

↑
n times

$$1 + 2 + 4 + 8 + \dots + \frac{n}{2} = n + n - 1 = O(n)$$

!!!

total cost

total cost

of an

array