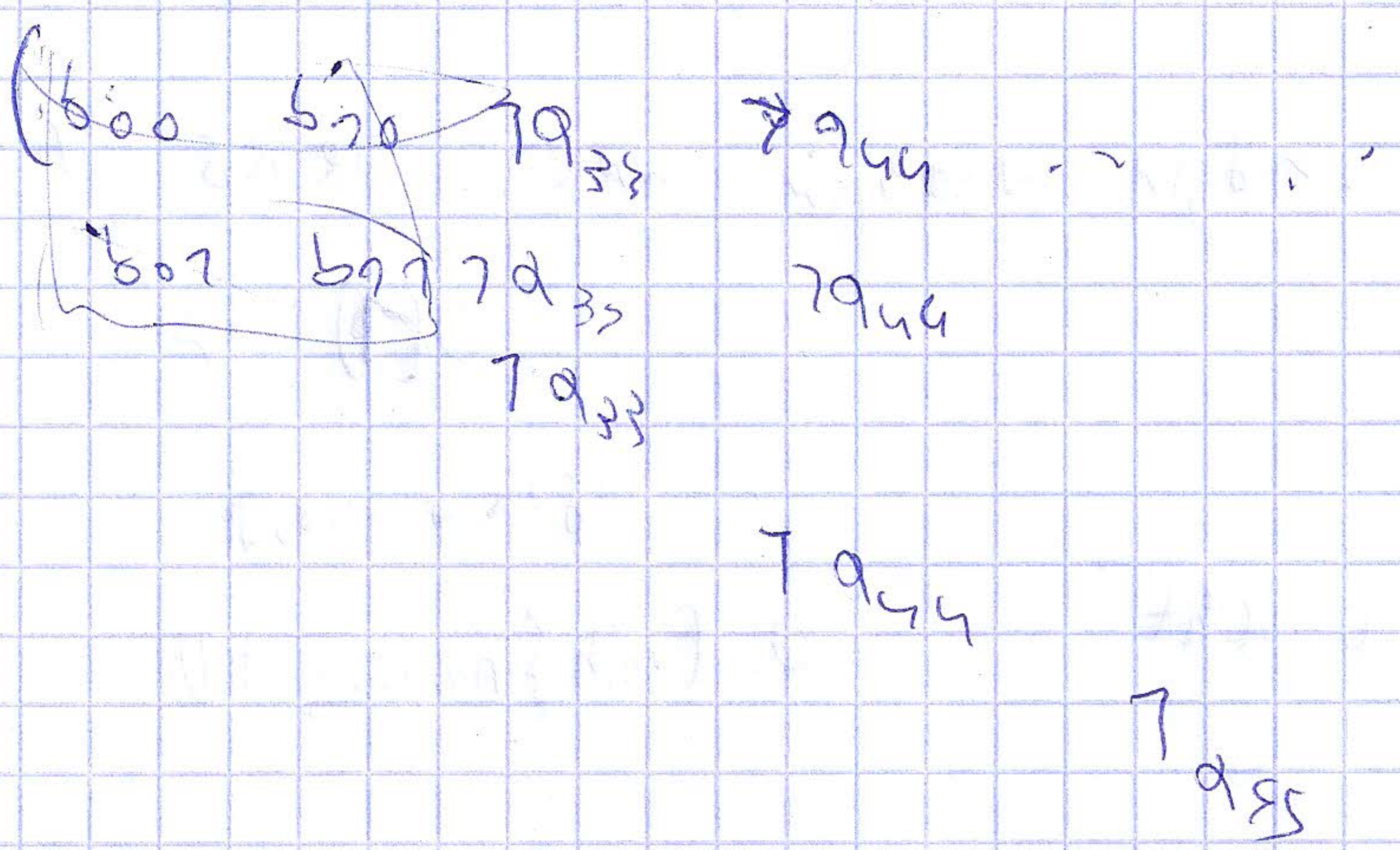


Q_{00} Q_{01}
 Q_{01} Q_{11}



2. Linked List

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

insert $O(n)$ delete $O(n)$

insert first $O(1)$
 delete first $O(1)$

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

insert first $O(1)$
 delete first $O(1)$

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

insert first $O(1)$ delete first $O(1)$

insert/delete first	$O(1)$	$O(1)$
insert last	$O(1)$	$O(n)$
delete last	$O(1)$	$O(n)$
Insert/Delete (i)	$O(\min\{i+1, n-i+1\})$	$O(i+1)$
Retrieve (i)	$O(1)$	$O(i+1)$

Concise notes

2

Max

Min

concat

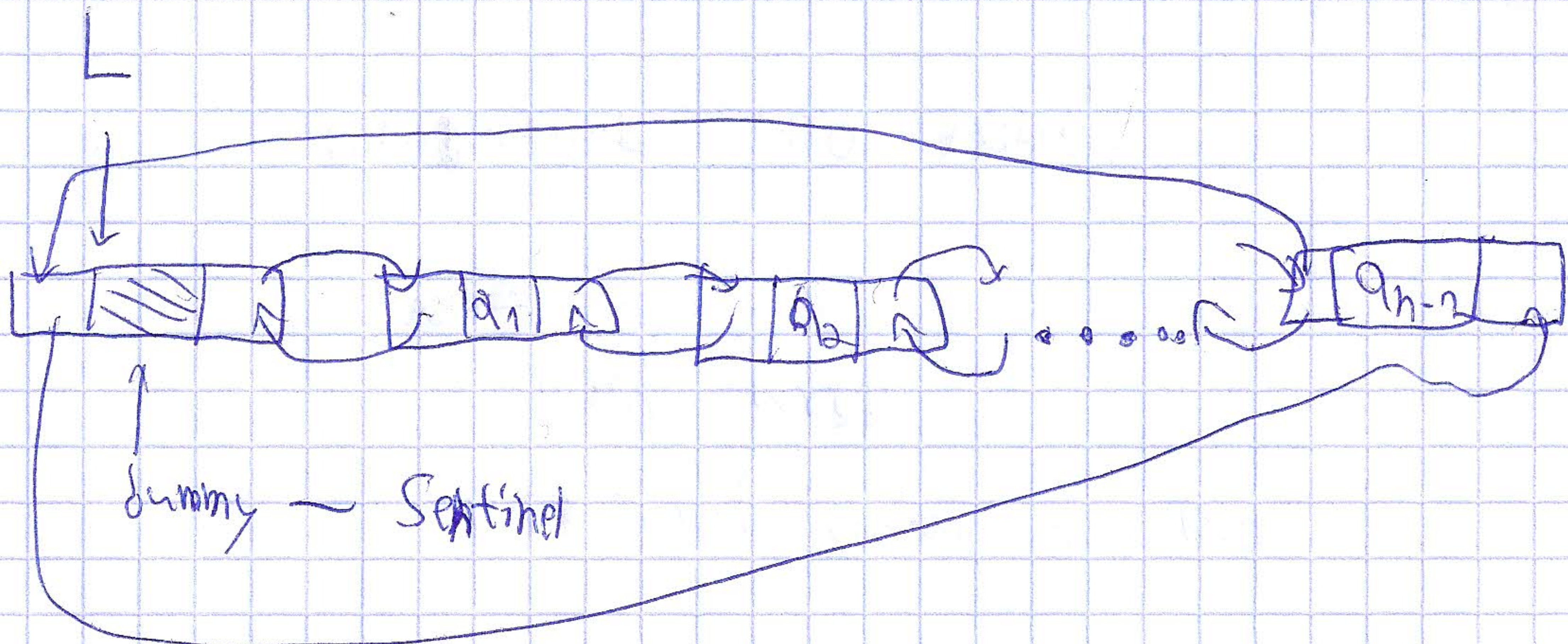
$O(\min\{h_1, h_2\} + 1)$

$O(n)$

first second

~~first second~~

doubly linked



Sentinel

first

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 $2n$.
 This operation takes $O(n)$ time.

Overall time complexity is $O(n)$.

For the average case, the time complexity is $O(n)$.

For the worst case, the time complexity is $O(n^2)$.

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

$\uparrow$
 n times

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

!!!

For the worst case, the time complexity is $O(n^2)$.

For the average case, the time complexity is $O(n)$.

For the best case, the time complexity is $O(1)$.

For the worst case, the time complexity is $O(n^2)$.