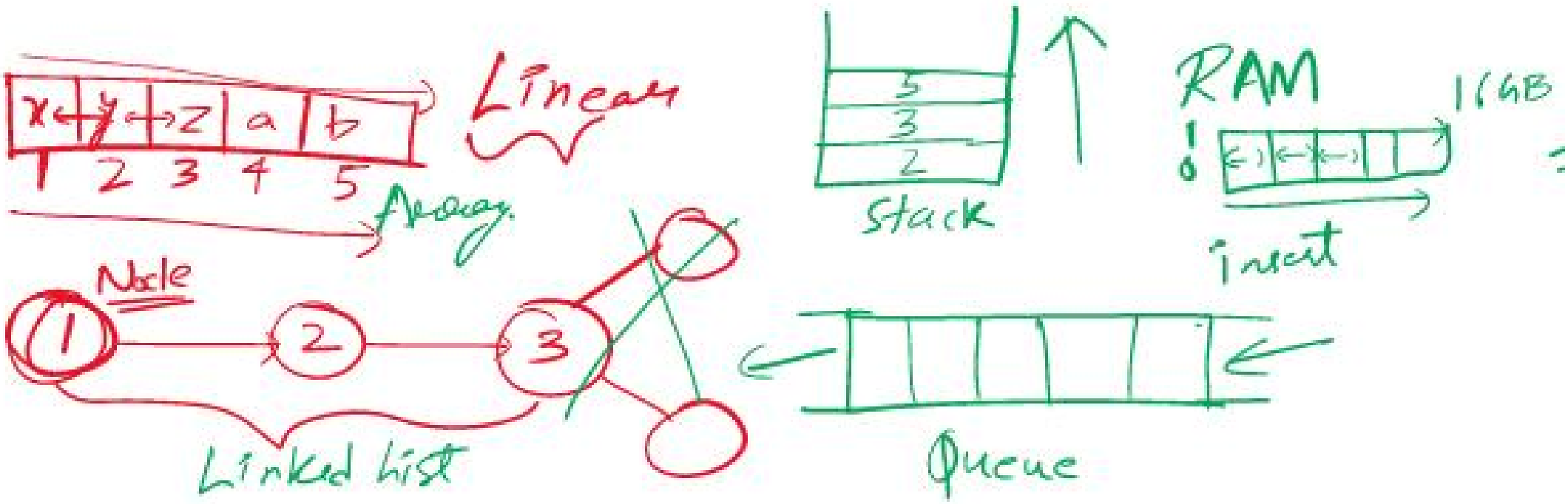
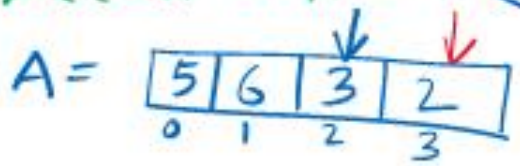

x
y
z
a
b
1 2 3 4 5
Array
Linear
5
3
2
Stack
RAM
16GB
Insert
Node
1
2
3
Linked list
Queue

Arrays - ADT

→ Fix size of elements

→ Same kind of elements

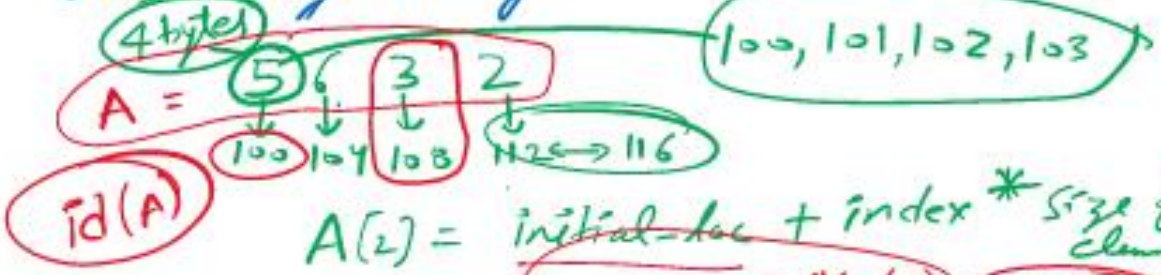
→ Access happens in O(1) time [Coz of Contiguous Storage]



A[2] = 3

A[1] =

How elements are stored in RAM when using Array.



A[2] = initial-loc + index * size of element

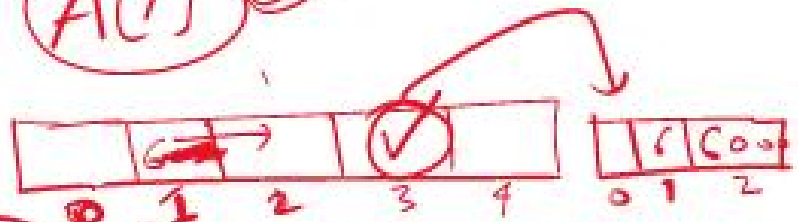
100 + 2 * 4 O(1)

= 108

100 + 101 * 4 = O(1)

for i = 0 to len(A) - 1

A[i]

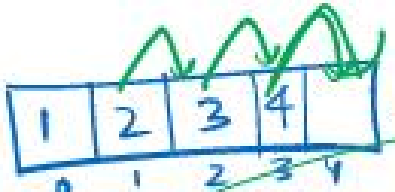


A[1] = 6

A[2] = 600

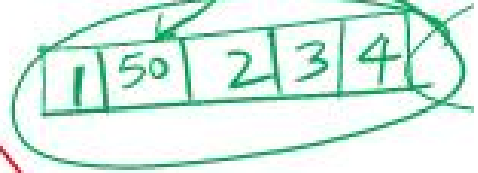
A[3] = initial + 3 * 4

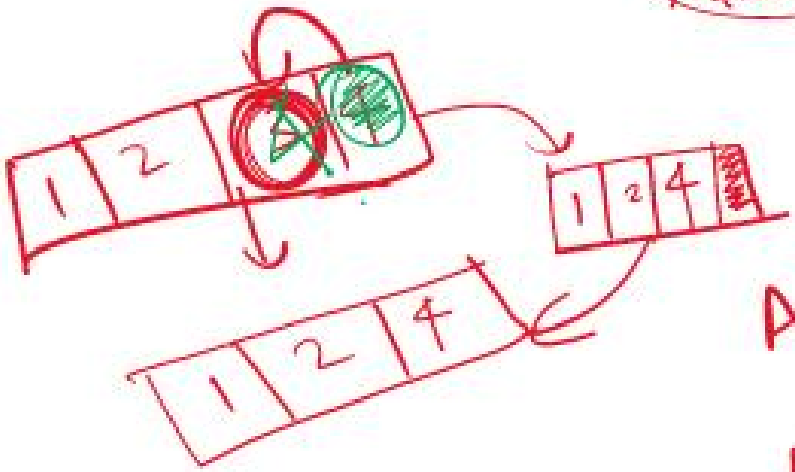
A =



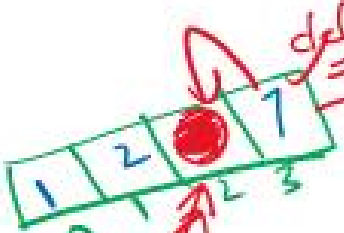
A[4] = A[3]

A.insertAt(1, 50)



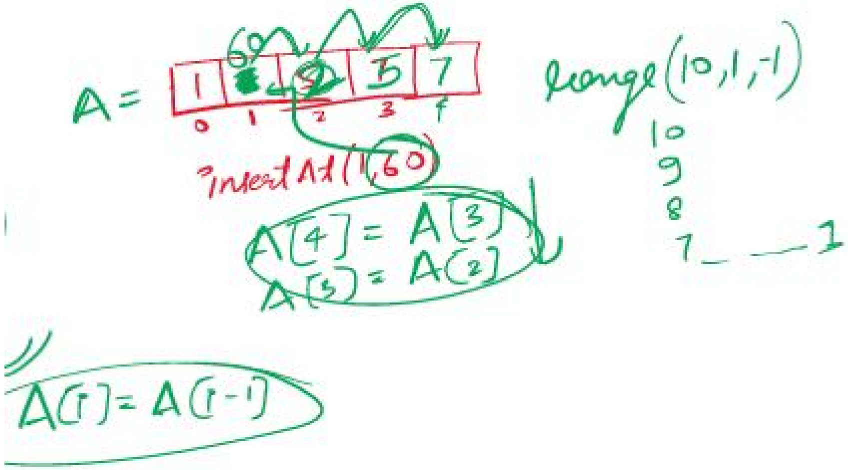


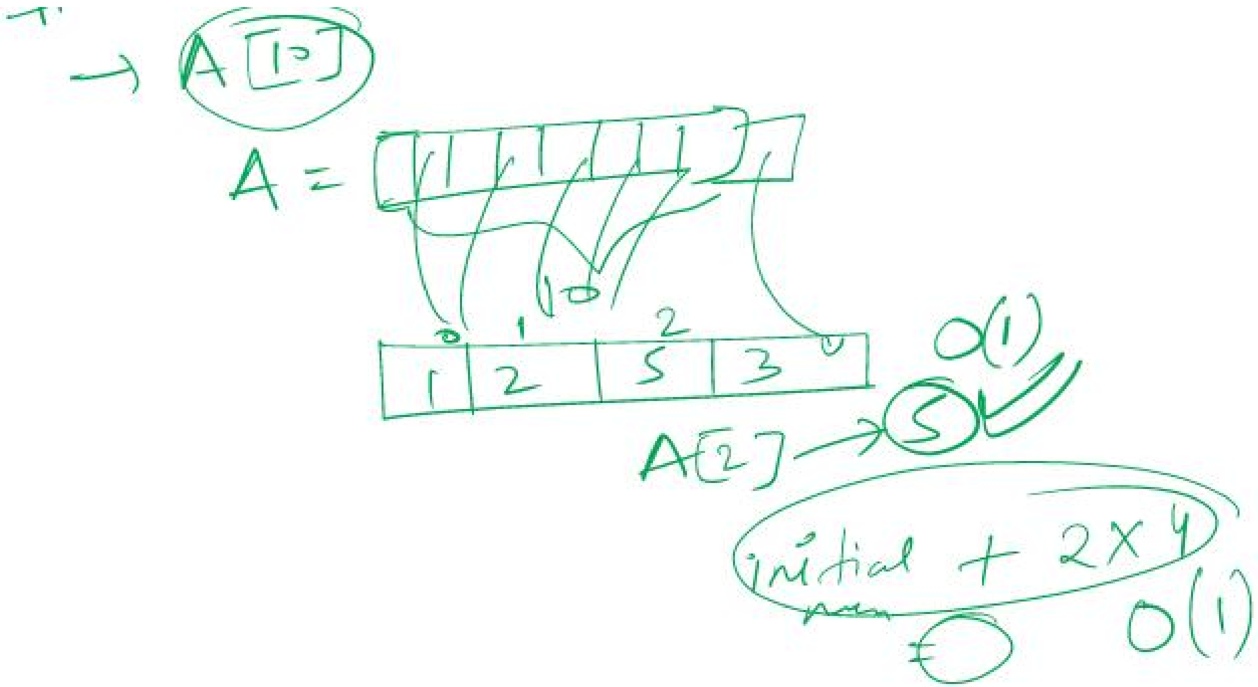
A =

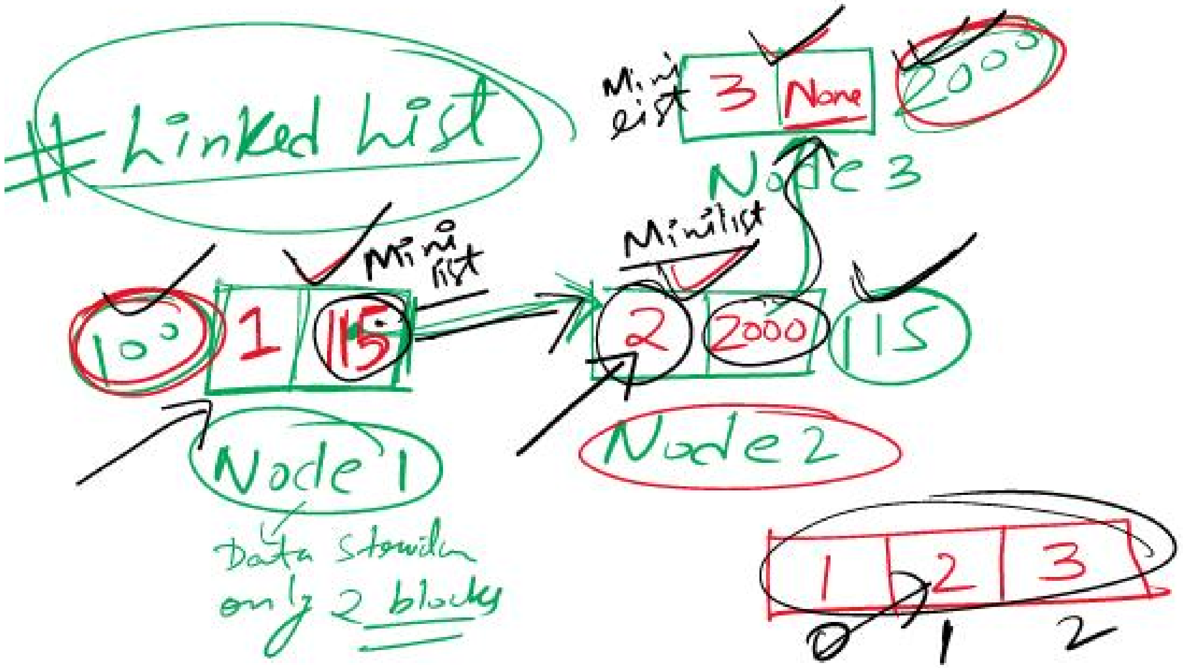


A[i] = A[i+1]

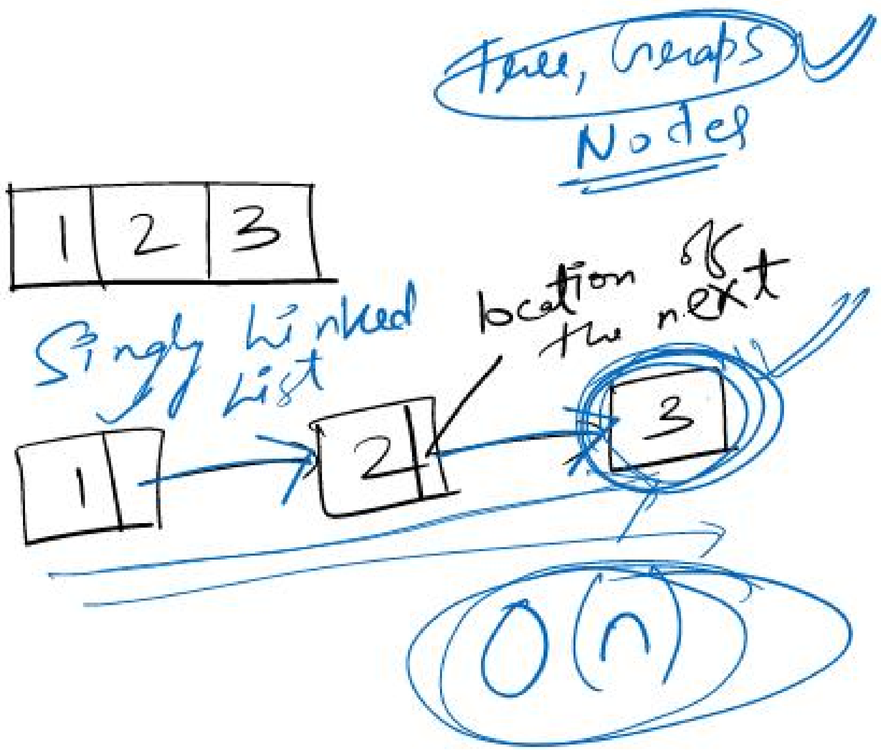
deleteAt(2)


A = 1 5 3 7
0 1 2 3 4
60
insert At(1,60)
A[4] = A[3]
A[3] = A[2]
A[i] = A[i-1]
range(10,1,-1)
10
9
8
7 — 1

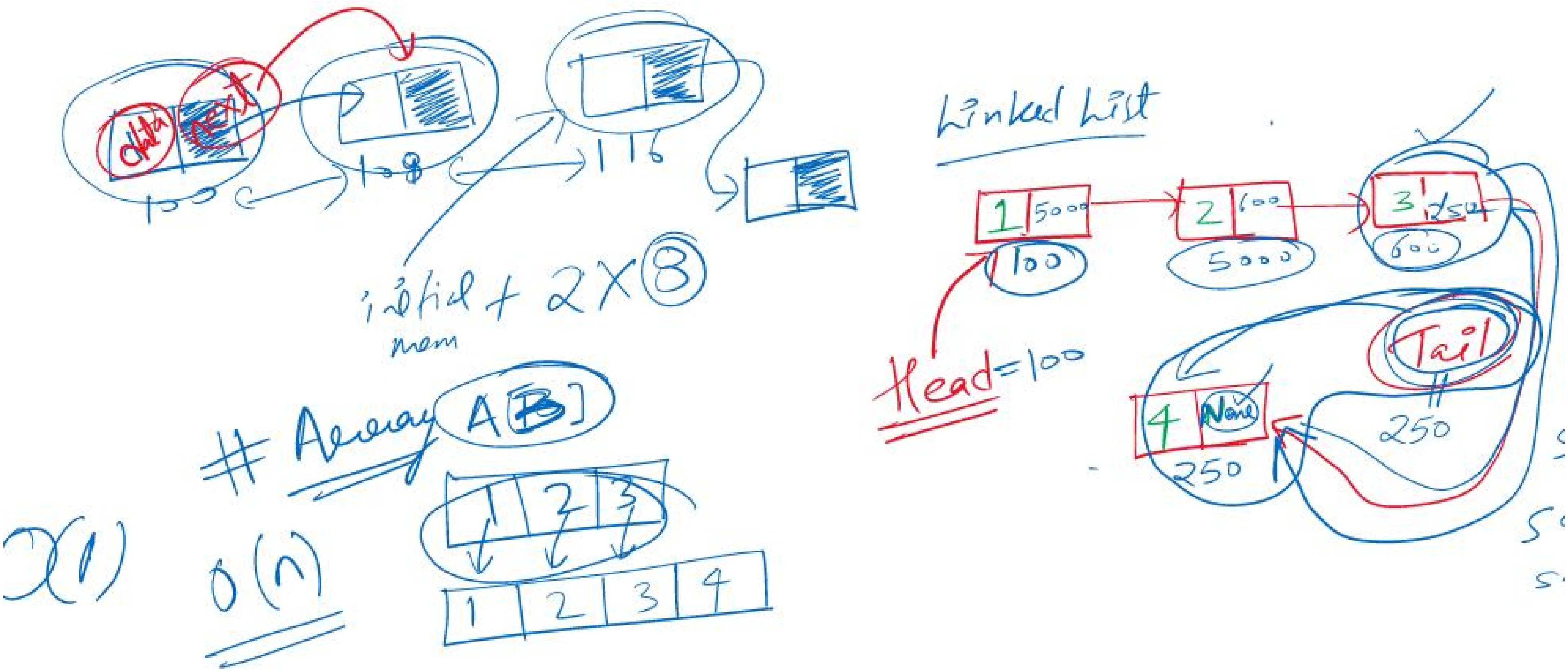

A[10]
A =
10
0 1 2
1 2 5 3
A[2] → 5
O(1)
initial + 2 X 4
O(1)



Linked List
Mini list
Node 1
Node 2
Node 3
None
2000
Minilist
Data store in only 2 blocks



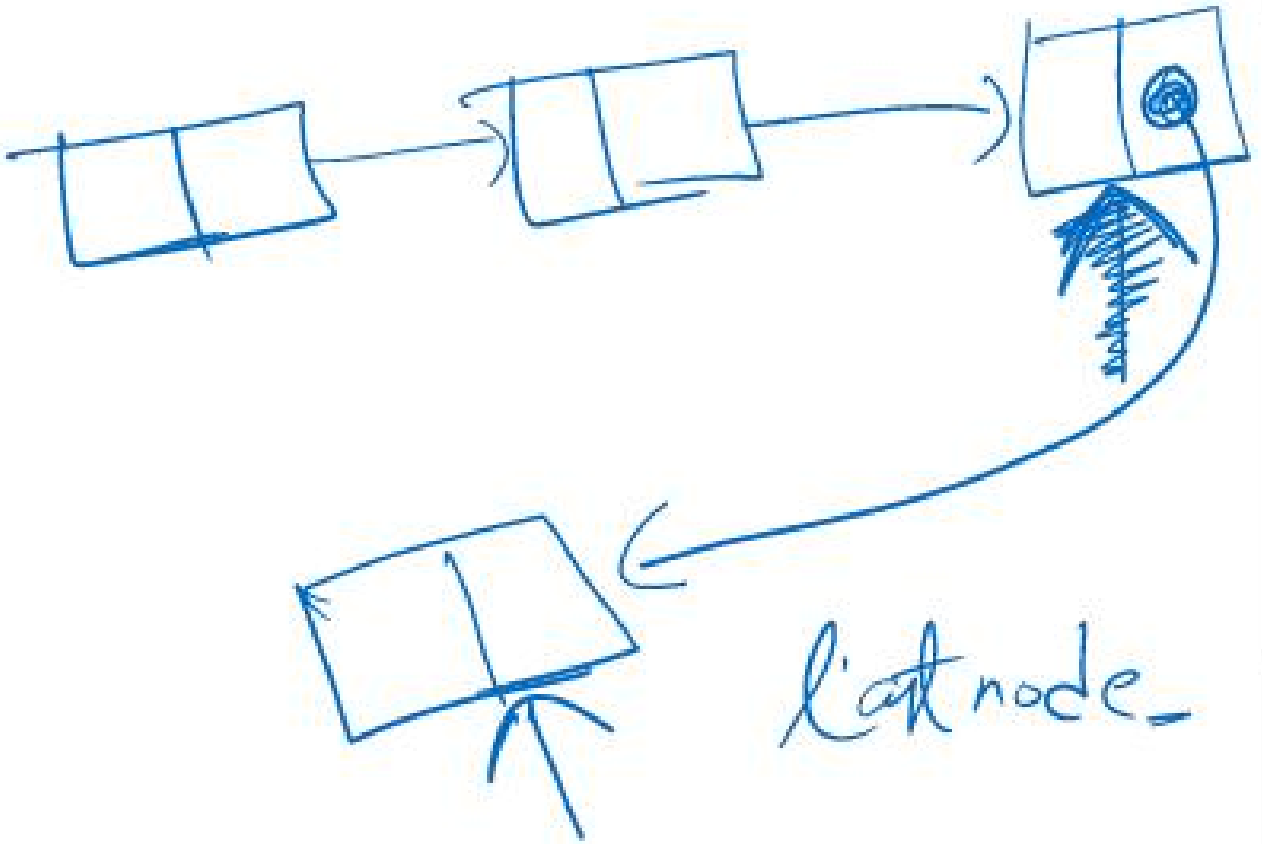
Tree, Graphs
Nodes
Singly Linked List
location of the next
O(n)



data
next
100
108
116
initial mem + 2X8
Array
A[B]
1 2 3
1 2 3 4
O(1)
O(n)
Linked List
1 5000
2 600
3 250
100
5000
600
Head=100
Tail
250
4 None
250

head = None

tail = None



self.tail = new_node

elf.tail.link = new_node

elf.tail = new_node