

CS 252

M, 1 April 2024
+ 3 April

Array lookup is $O(1)$

int a[100];

4 bytes →

address
of the
beginning
of the
array

$a[73] = ?$

$a + 4 * 73$

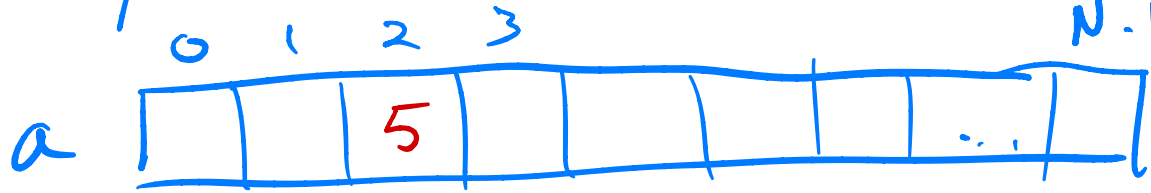
→ address of

$a[73]$

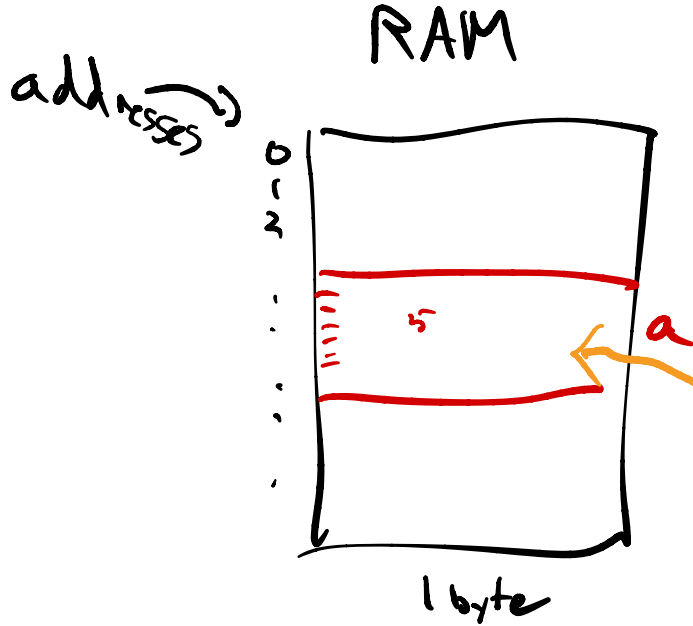
"RAM" = "random-access memory" = cont

RAM ~ random access
means — it takes the
same time to get
any byte @ any address

Arrays



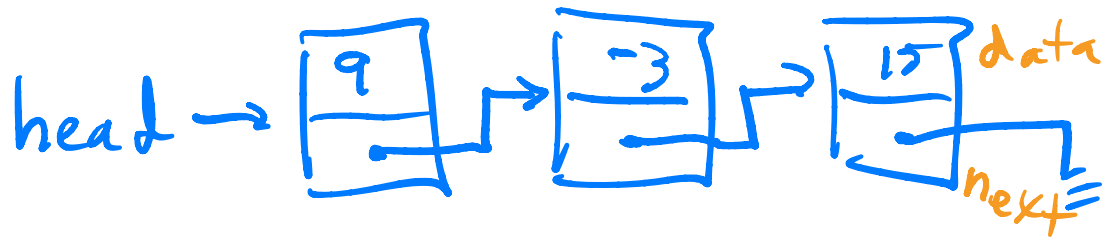
```
int a[N];  
a[2]=5;
```



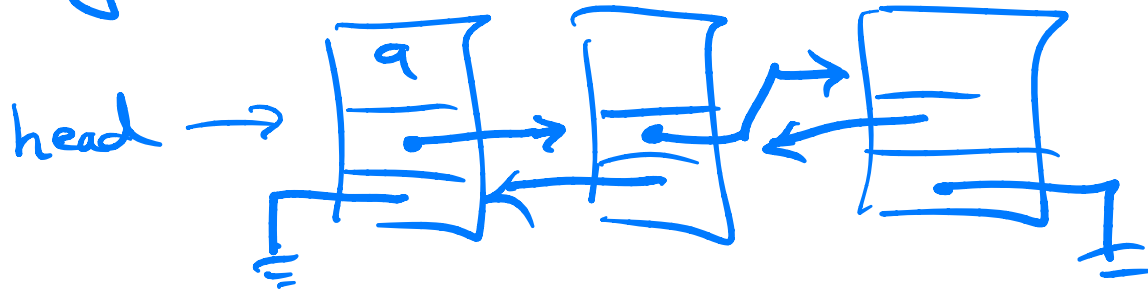
An array occupies a contiguous block of memory

$a[9]$ address
is $a + 4 * 9$
int index

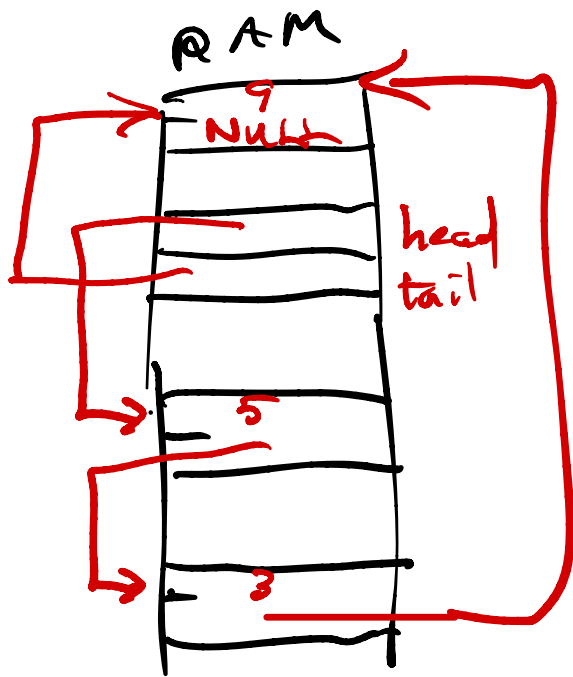
Linked list



Doubly linked



w/ tail ptr (singly-linked w/ tail ptr)



"List interface" ordered seq of items

any object that supports
these ops

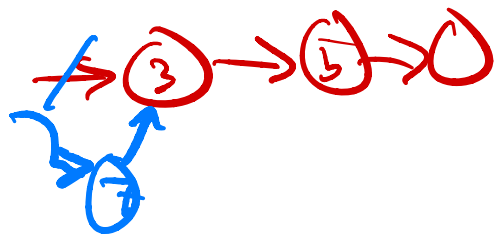
- get ~~the~~ item at index k
- set item " " "
- insert item before index k
- etc.

List (either linked or array)

Insert ⁷ at beginning

Array impl. of List: $O(N)$
shift everything down 1

Linked list? $O(1)$



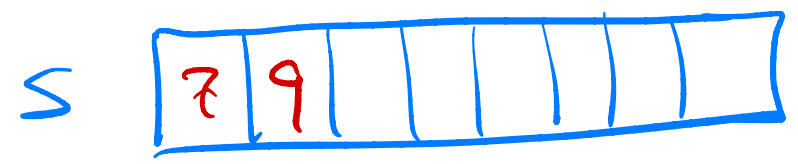
`newnode = new Node(7)`

`newnode.next = head`

`head = newnode`

Stack w/ array

top -1



empty
~ top == -1

push(7, s)

top++

s[top] = 7

push(9, s)

O(1)

pop(s)

save = s[top]

top--

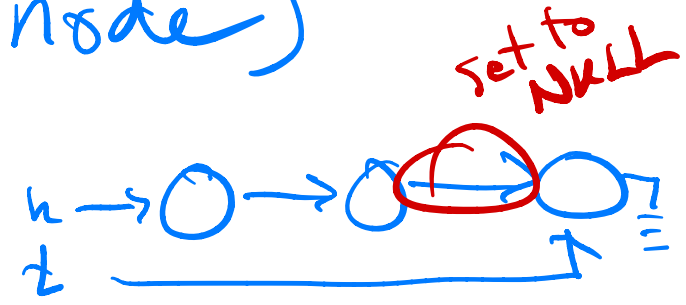
return save

Stack w/ LL

top = head of list
(1st node)

push $O(1)$

pop $O(1)$



delete from tail

top = tail of list — bad

Queue w/ LL w/ tail ptr

Front = head LL

End of \downarrow
= tail LL

Add to tail $O(1)$

Delete from head $O(1)$