CS 252

M, 1 April 2024

+3 April
Array lookup is $O(1)$

```cpp
int a[100];

a[73] = ?

a + 4*73

$\Rightarrow$ address of $a[73]$

"RAM" = "random-access memory" = $\text{co}$
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.

\[ a[9] \text{ address is } a + 4 \times 9 \]
Linked list

head - 9 -> -3 -> 15

Doubly linked

head - 9 -> -3 -> 15

data

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

head → 5 → -3 → 9

RAM

head

tail

null

3

5
“List interface” ordered seq of items any object that supports these ops
- get 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

\[ \text{top} \begin{array}{c} 7 \\ 6 \end{array} \]

empty \iff \text{top} = -1

\begin{align*}
\text{push}(7, s) & : \quad \text{top++} \\
& : \quad s[\text{top}] = 7
\end{align*}

\begin{align*}
\text{push}(9, s) & : \quad \text{O}(1)
\end{align*}

\begin{align*}
\text{pop}(s) & : \quad \text{save} = s[\text{top}] \\
& : \quad \text{top--}
\end{align*}

\text{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 = tail LL

Add to tail O(1)
Delete from head O(1)