Announcements & Logistics

A1 due Friday Oct 19.
Midterm Tuesday Oct 23.

**No prep** for next week. (Use the extra time for studying!)
From last time, `LinkedList.__contains__`

We care about running time as a function of input size:

- “constant”  \(O(1)\)
- “linear”  \(O(n)\)
- “quadratic”  \(O(n^2)\)
From last time, LinkedList.__contains__

def __contains__(self, item: Any) -> bool:
    curr = self._first
    while curr is not None:
        if curr.item == item:
            return True
    return False
From last time, `LinkedList.__contains__`

Running time can vary, even for a fixed input size!

We’ll revisit this idea later in the course.
Recursion

CSC148, INTRODUCTION TO COMPUTER SCIENCE
DAVID LIU
Data structure informs code structure

```python
i = 0
while i < len(lst):
    ... lst[i] ...
i += 1

curr = lst._first
while curr is not None:
    ... curr.item ...
curr = curr.next
```
Data structure informs code structure

```
List[int]

for x in lst:
    ... x ...

List[
    List[int]
]

for lst in lst_of_lsts:
    for x in lst:
        ... x ...
```
Data structure informs code structure

```python
List[
    List[
        List[int]
    ]
]  
for lst_of_lsts in lst_of_lsts_of_lsts:
    for lst in lst_of_lsts:
        for x in lst:
            ... x ...
```
Data structure informs code structure

A nested list is...

- An integer
- A list of nested lists

```python
def nested_f(obj):
    if isinstance(obj, int):
        ...
    else:
        for sublist in obj:
            ... nested_f(sublist) ...
```
Partial tracing practice

Attempting to fully trace recursive code is time-consuming and error prone.

*When tracing recursive code, don’t trace into recursive calls! Instead, assume each call is correct, and make sure the rest of the code uses those calls correctly.*