February 3, 2025

LEC 11 - Linked Lists

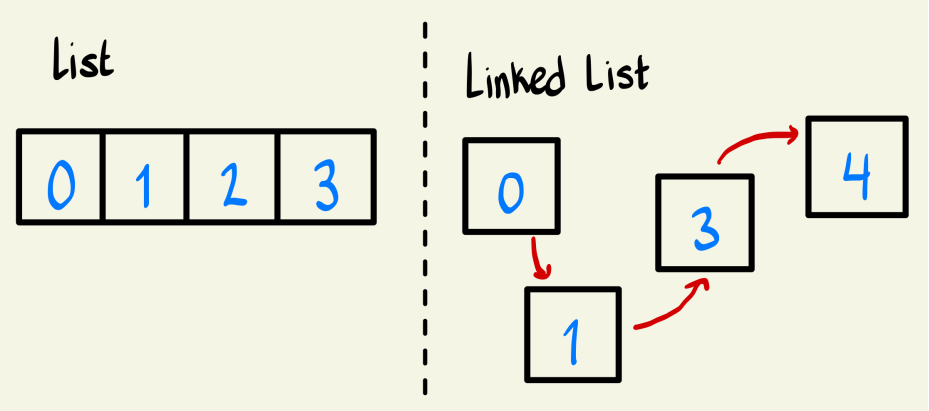
Types of Lists

Array-Based

- Store references to element in contiguous blocks
- Constant indexing
- Allocate large blocks of memory, which becomes increasingly difficult as memory is in use
- Overkill in some situations

Linked

- Store elements anywhere, but each element must also store a reference to the next item in the list
- Each element stores a reference to the next element
- Reserve just enough memory for the object value they refer to, and a reference to the next element



Linked Lists

- There are useful ways to thinking about linked list nodes
    1. As a list made up of an item and a sub-list
    2. As objects (nodes), each containing a value and a reference to another similar node object (the next “link”)

Data Structures

```
class _Node:
      item: Any
      next: Optional[_Node]

class LinkedList:
      _first: Optional[_Node]
```

February 3, 2025

LEC 11 - Linked Lists

Traversing Linked Lists

- Make a reference to at least one node, and movie it along the list

```
curr = self._first
while curr is not None:
      curr = curr.nex
```