Today, a new data structure! Linked list!!

Arrays suffer from 2 major drawbacks.

1. They have a maximum size
2. While insertions and deletions at the front or back can be O(1), operations in the middle can’t

Linked lists were designed to overcome these drawbacks.

A linked list consists of a sequence of nodes. Each node holds data and a reference to the next node of the list.

Each node in a linked list is individually allocated so the list can grow without bound and you only “pay” for what you need.

Special list terminology

* Single linked
* Doubly linked

A doubly linked list lets us move both forwards and backwards but requires extra space.

In circular list, the last node has a reference to the first node, and for a doubly linked list, the first node has a reference to the last node as well

What would happen if we tried to use binary search on a sorted linked list?

It would not be practically. Why? Unlick arrays which support indexing in constant time, linked lists do not. To reach to ith node in a list requires time proportional to i.