Implementing Algorithms for Beginners

Linked List Example:

Linked List 🡪 a list of objects that point to the next object

Define a Single Case:

\*Single example of an algorithm allows you to visualize & reduce it to the General Case\*

Single Case 🡪 inserting a node at specific position

* Often in a linked list you want to **insert a node** into any arbitrary position in the array
* In order to insert a new node 🡪
* Must find the position you want to insert the node at – 1 (AKA the node prior to that position)
* Have the node prior in the list point to the new node
* Have the new node point to the node prior was originally pointing at

Define the General Case:

\*General Case 🡪 Inserting any node N at position i

* Whichever position you want to insert at (i), you need access to the Prior-Node – right in front of it (P)
* Set the new node (N) ‘s next property to point at the After-Node – the same node P is currently pointing at

\*(MUST DO THIS STEP FIRST OR ELSE YOU LOSE THE REFERENCE TO THE AFTER-NODE)\*

* Then set P’s next node property to point at the new node N

\*Subroutines 🡪 mini algorithms that are part of the bigger algorithm

Define the Edge Cases: special cases that don’t apply to the general algorithm

Edge Cases for Linked List –

* Inserting a Node at the beginning of the list (at position 0)
* Inserting a node at the end of the list

Merge Sort 🡪 Recursion

Recursive Methods 🡪

* A method that calls itself
* With each method call, the problem becomes more and more solved
* Eventually reaches a point where the method no longer makes any more method calls to itself

Triangular numbers???