

# Find middle node

① 1 2 3 4

length = 5

$$\text{middle} = 5/2$$

$$= \text{floor}(2.5)$$

$$= 2$$

mid is at index

In the case of two middle  
take the second

① 1 2 3 4 5

$l = 6$

$$m = 6/2 = 3 \text{ index}$$

## Improved algorithm

The previous approach needed two loops to get the middle node.

There's an alternative by having two variables, the first variable will walk the list one by one, this will be called **slow**. The other will walk twice, 2 steps at a time, this will be called **fast**.

**slow fast**

① 1 2 3 4 5

slow and fast start at head

**slow fast**

① 1 2 3 4 5

**slow**

**fast**

① 1 2 3 4 5

**slow**

**fast**  
NULL

When **fast** reaches **null** it means we are done with it. **Slow** is now the middle item, this happens because **fast** advances two by two, whereas **slow** goes one by one. To achieve this the **while** loop is conditioned to **fast** and **fast->next** to be not **NULL**.

■ **NOTE**: **fast** doesn't advance one more from the **slow**, instead it always advance two from itself.