

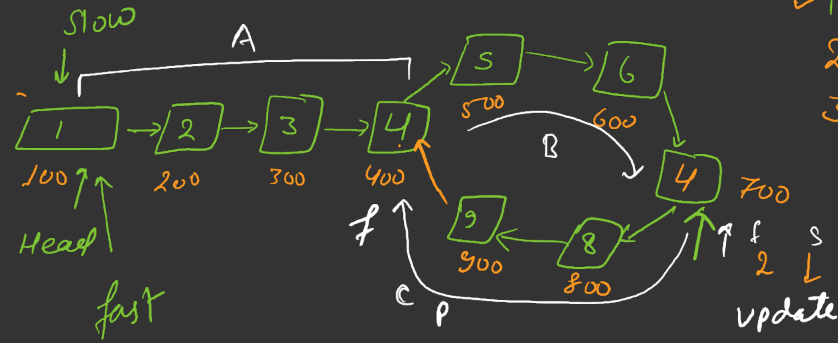
## Remove Loop in Linked List

✓ 1-loop detect

2 - slow - head

3 - starting of loop

$$A = B + C + C$$



$$\text{Slow} : A + (B + C) \times i + B$$

$$\text{Fast} : A + (B + C) \times j + B$$

$$\text{Fast} = 2 \times \text{Slow}$$

$$j = \text{fast}$$

$$i = \text{slow}$$

$$j > 2i$$

$$A + (B + C) \times j + B = 2A + 2(B + C) \times i + 2B$$

$$A + B = (B + C)j - 2(B + C)i$$

$$= (B + C) \times (j - 2i)$$

$$A + B = 2(B + C) \quad x=1$$

$$A + B = B + C \quad | \quad A = C$$

$$A + B = 2B + 2C$$

$$A = (B + C) + C$$

Node \* slow = head, \* fast = head;

while (fast != NULL && fast->next != NULL) {

slow = slow->next;

fast = fast->next->next;

if (slow == fast)

break;

}

if (fast == NULL || fast->next == NULL) {  
return; }

slow = head; Node \*prev = ~~NULL~~ fast;

while (slow != fast) {

slow = slow->next; prev = fast;

fast = fast->next;

}

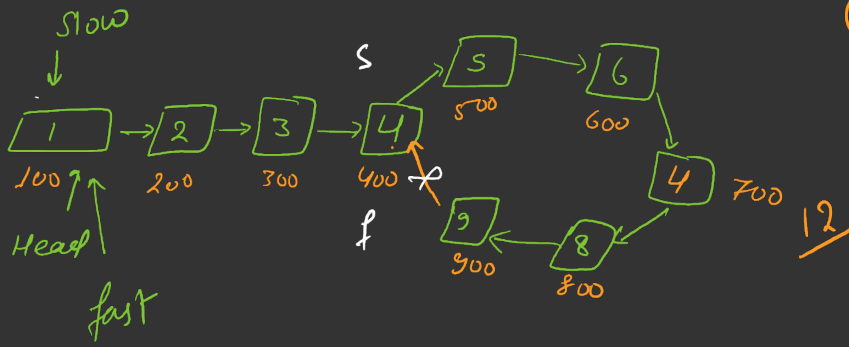
prev->next = NULL;



$$S = h$$

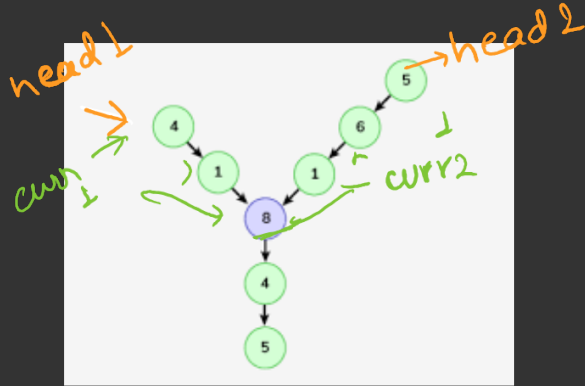
prev = fast

prev  $\rightarrow$  next = NULL  $\times$



① Loop detected  
6-length

## Intersection in Y Shaped Lists



Node \* curr1 = head1, \* curr2 = head2;

int count1 = 0, count2 = 0;

```
while (curr1) {  
    count1++;  
    curr1 = curr1->next;  
}
```

```
while (curr2) {  
    count2++;  
    curr2 = curr2->next;  
}
```

curr1 = head, curr2 = head;

```
while (count1 > count2) {
```

```
    curr1 = curr1->next;  
    count1--;  
}
```

```
while (count2 > count1) {
```

```
    curr2 = curr2->next;
```

```
    count2--;
```

count1 = 5  
count2 = 6

```
while (curr1 != curr2) {
```

```
    curr1 = curr1->next;
```

```
    curr2 = curr2->next;
```

```
}
```

```
if (!curr1)  
    return -1;
```

```
return curr->data
```













