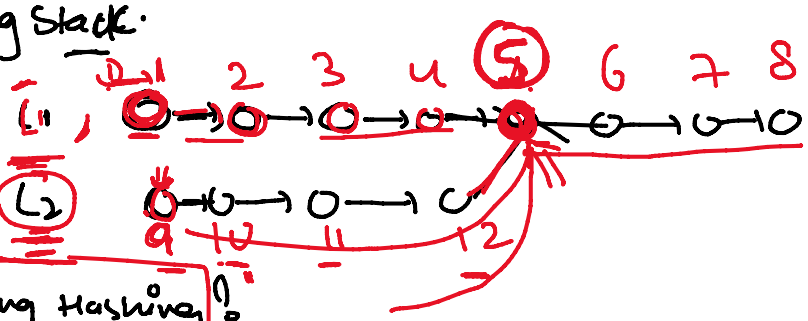
Class 52
06 April 2023 21:07

CLASS-52

Intersection of 2 linked list

→ Brute force

→ Using Stack.

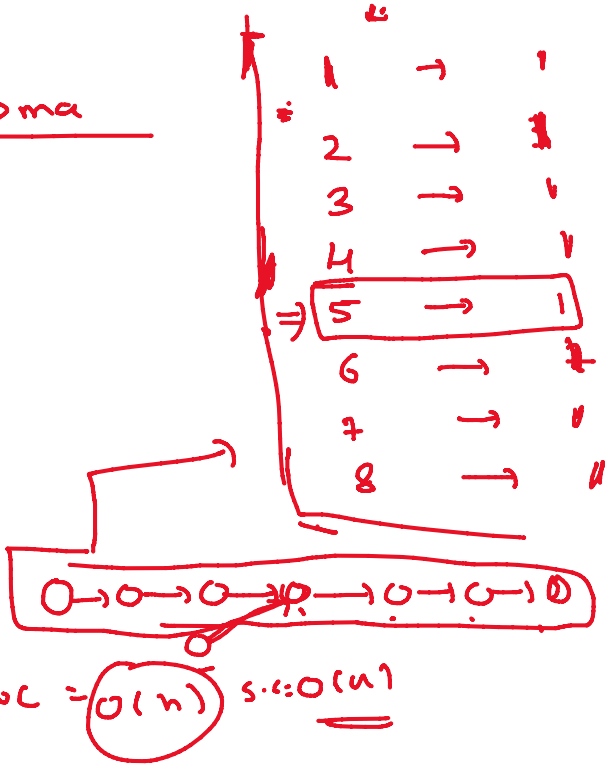


→ Using Hashing !!

Maps ⇒ store, map < Node*, int > ma

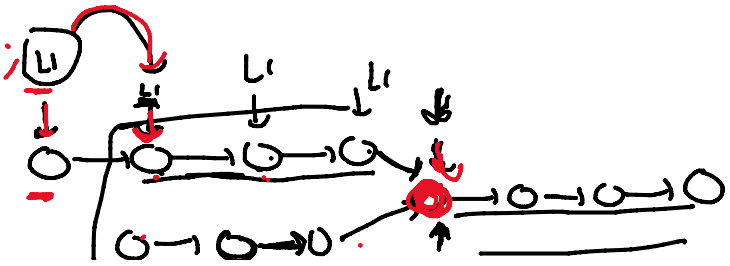
⇒ They both are intersecting

```
unordered_map< Node*, int > ma;

while ( head1 != NULL )
{   ma.insert( head1 );
    head1 = head1 -> next ;
}

while ( head2 != NULL )
{   if (ma.count(head2))
    {   cout << "Found" ;  return ;
    }
    head2 = head2 -> next ;
}
cout << "-1"
return NULL;
}
```

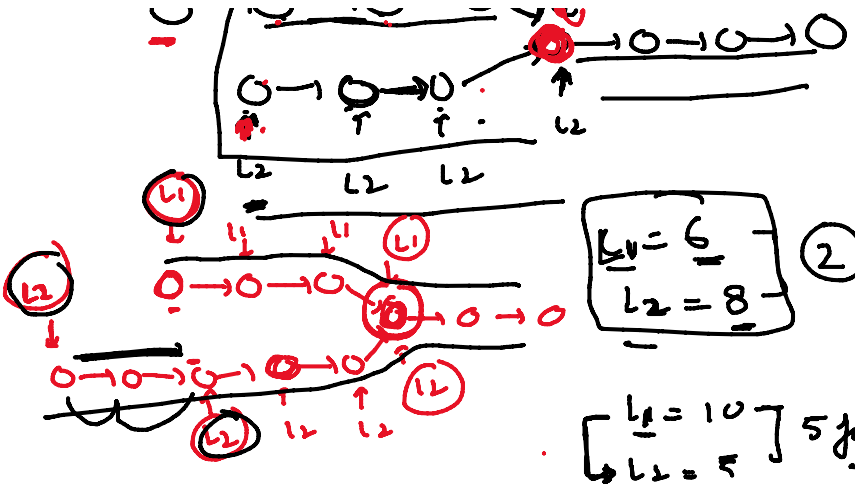


ToC = O(n) S.C: O(n)

4th Approach

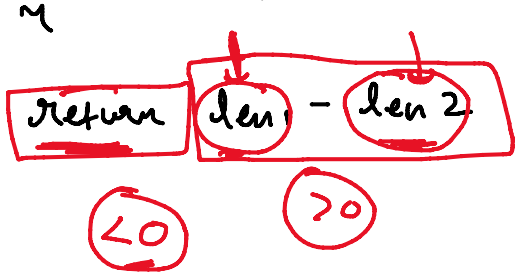


4th Approach




```
Node* l1;
Node* l2

while( l1 and l2 )
{   if (l1 == l2)
    { return l1; }

    l1 = l1 -> next ;
    l2 = l2 -> next ;
}

return NULL
```

```
int getdiff ( Node* l1 , Node* l2)
{ // return diff b/w length of
    l1 and l2
  int len1 = 0;
  while ( l1 != NULL )
  {   l1 = l1 -> next;
      len1 ++;
  }
  int len2 = 0;
  while ( l2 != NULL )
  { l2 = l2 -> next;
    len2 ++;
  }
  return len1 - len2
}
```




```
void intersection ( Node* h1 , Node* h2)
{
  int dif = getdif ( h1 , h2 );
   if ( dif < 0 )
  { while ( dif++ != 0 )
      h2 = h2 -> next ;
  }
```

5 - (7) = (-2) → 0

(2)

For (

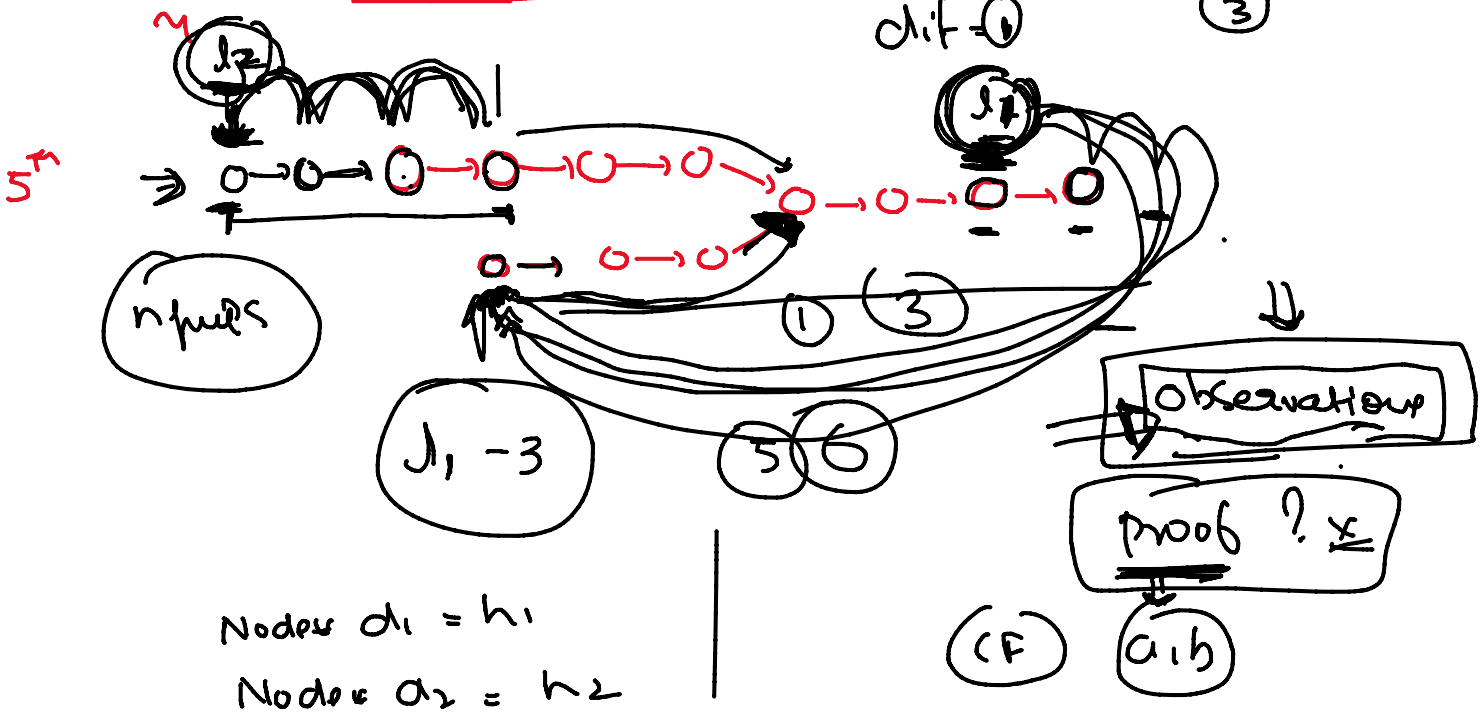
else

while (dif-- != 0)

head1 = head1 → next ;

dif = 2

l1 = 7 - l2 = 5 = 2



Node* d1 = h1

Node* d2 = h2

bool flag1=0, flag2=0;

while (d1 != d2)

{ if (d1 == NULL) → if (flag) return;

{ d1 = head2; flag1 = 1; }

else

d1 = d1 → next ;

if (d2 == NULL) → if (flag2) return;

{ d2 = head1; flag = 2;

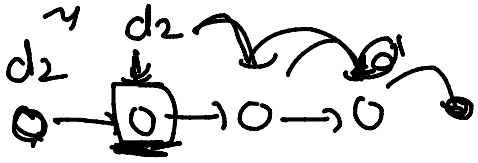
}

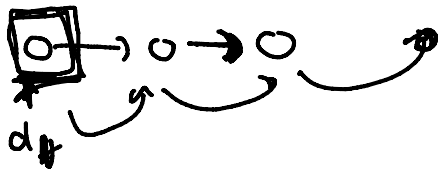
else

{ d1 = d2 → next ;

}

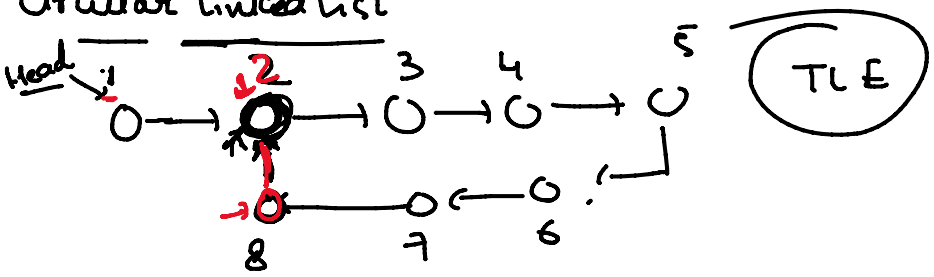
}



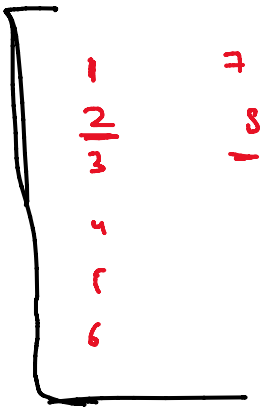


→ Detect Loop in a linked list

Circular linked list



TLE



→ Detect ? map < Node*, int > ma.

Node* head;

while (head != NULL)

{ if (ma.count(head))

{ cout << " " << "; return; }

ma[head]++;

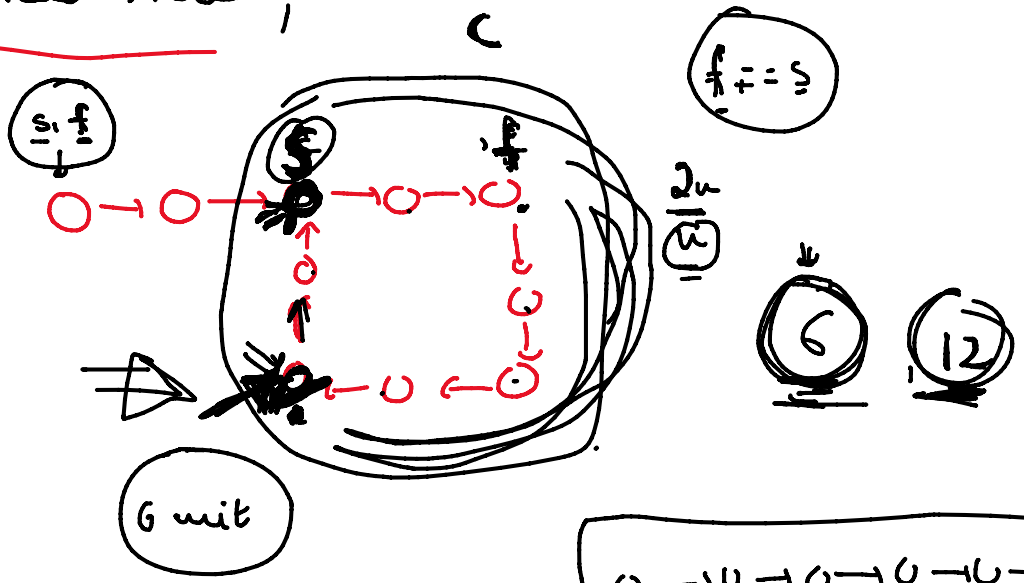
head = head → next;

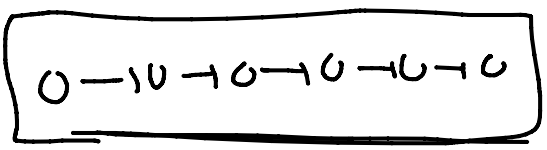
}

→ 2nd Approach

$s = k \cdot L + a$

$f = 2 \cdot k + a$

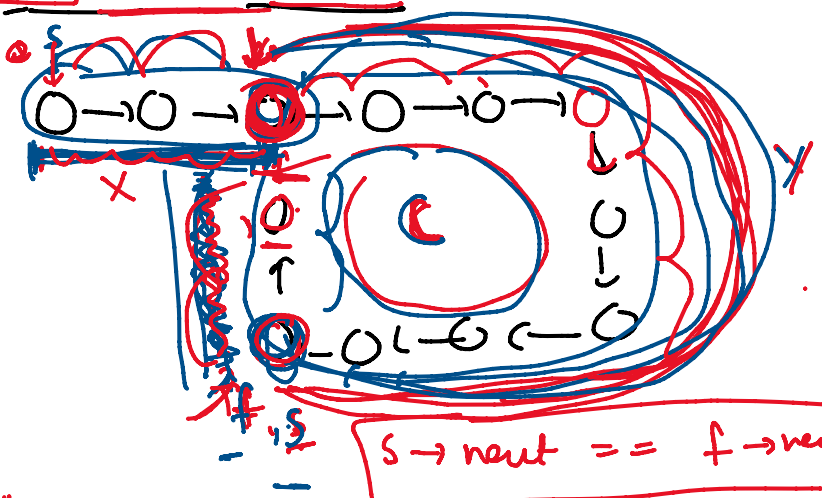




bool cycleDetect (node* head)

```
{   if (Head == NULL) return false;
    node* fast = Head;
    node* slow = Head;
    while ( fast != NULL and fast->next->next != NULL)
    {   fast = fast->next->next;
        slow = slow->next;
        if ( fast == slow)
        {  // cout << "cycle exist";
           return;
        }
    }
    cout << "Single linked list";
```

→ Remove loop in a linked list



(1.1)

$x = (l \cdot c) - y$

$S = x + k \cdot c + y$

$F = x + 2 \cdot c + y$

$F = 2 \cdot S$

$x + 2 \cdot c + y = 2 (x + k \cdot c + y)$

$+ 2 \cdot c = x + y + 2k \cdot c$

(6) (12)

$+ \quad 2 \cdot C \quad = \quad x + \quad y + 2k \cdot C$

$2 \cdot C = x + y + 2k \cdot C$

$x + y = 2 \cdot C - 2k \cdot C$

$x + y = (2 - 2k) \cdot C$

L

$x + y = L \cdot C$

$x = (L \cdot C) - y$

```
Slow = Head;
while ( slow → next != fast → next )
{   Slow = Slow → next;
    fast = fast → next;
}
fast → next = NULL;
```