Todays Contest

1. Add 2 linked lists
2. Find duplicate number
3.

208. You are given 2 linked lists.

Each node contains a single non-negative digit.

Add both linked lists & return head node.

Ex1:

h1: 3 → 4 → 8 → 6 → NULL

h2: 9 → 8 → 7 → 4 → 5 → NULL

Idea:

```
Node* Add(Node *h1, Node *h2){

}
```

28 Find the Duplicate Number

Given an array of Integers containing N+1 Integers.
Where each Integer is in range [1..N].
There is only 1 repeated number in nums, return this repeated number

Note: cannot modify array & no extra space.

Ex 1: nums[5] = {1 3 4 2 2} ans = 2
N+1 = 5
N = 4

Ex 2: nums[5] = {3 1 3 4 2} ans = 3
N+1 = 5
N = 4

Ex 3: nums[5] = {3 3 3 3 3} ans = 3
N+1 = 5
N = 4

Idea 1:

```
Iterate on arr[];
    For arr[i]; Iterate on arr[] & calculate frequency = c;
    if ( c > 1) {
        return arr[i];
    }
}
```

TC: O(N^2) SC: O(1)

Ideas:

nums[5] = {1 3 4 2 2} → {1   4}

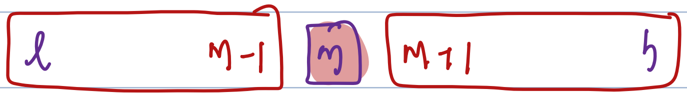
Target: Repeate element

Search Space: In arr[] *

We cannot discard searchspace, change it

Target: Repeated element

Search Space: $l = 1$, $h = N$

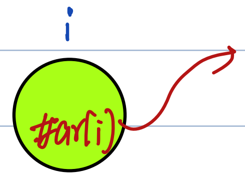


Using BS

```
            0  1  2  3  4  5   6   7  8  9  10 11 12 13  14  15
nums[16] = {4  7  7  8  7  14  15  5  2  1  3  7  7  12  11  10}
# N+1=16
  N=15

l    h    m
1    15   8    #Iterate & calculate ele i=8 : 11
               {1.. 8} ; ans=m h=m-1;    {9..15}
               dis = 8 ✓   #ans=8           *
               ele = 11

1    7    4    #Iterate & calculate ele i=4 : 4
               {1..4}  {5...15}
               dist=4   dis= 11
               ele=4    ele =12
               l=m+1;

5    7    6    #Iterate & calculate ele i=6 : 5
               {1.. 6}  {7..15}
               dist=6   dis= 9
               ele= 5   ele= 11
               l=m+1

7    7    7    #Iterate & calculate ele i= 7 : 10
               {1.. 7}
               dist=7   ans=7; h=m-1;
               ele=10

7    6    # break & return ans = 7.
```

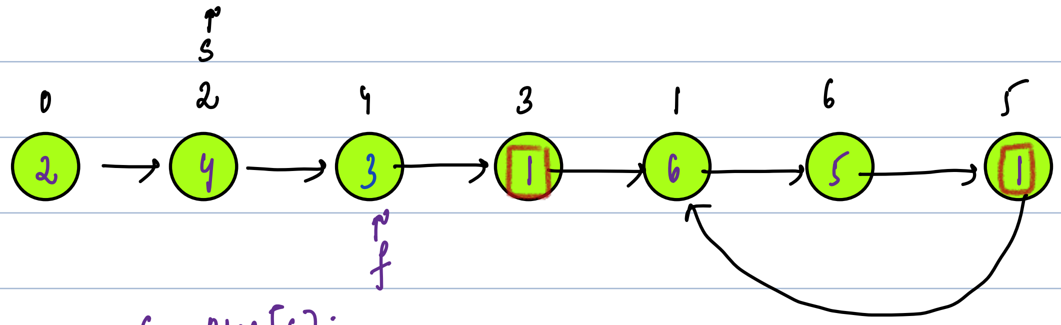
#Idea3: Map to linked list.

#ar[i]



Ex3:

```
            0  1  2  3  4  5  6
num[7] = { 2  6  4  1  3  1  5 }
```

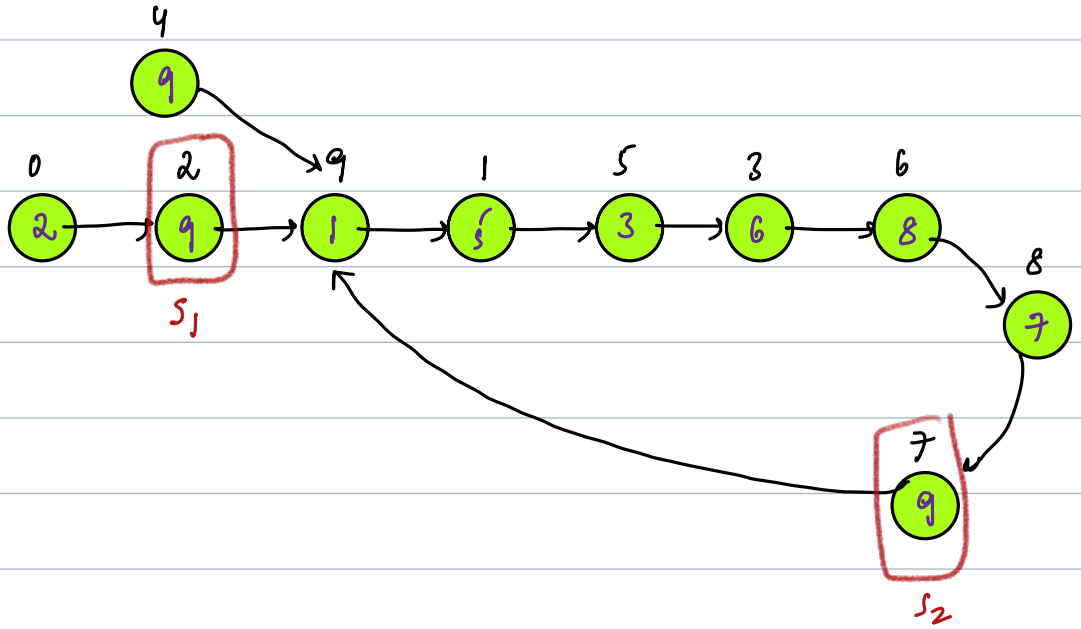


s = num[s];

f = num[num[f]]

Ex4:

```
             0  1  2  3  4  5  6  7  8  9
num[10] = { 2  5  9  6  9  3  8  9  7  1 }
```



```
int findDuplicate(vector<int> &nums){ TC: O(N)  SC: O(1)

    int s=0, f=0;
    while(1){
        s= nums[s];
        f= nums[nums[f]];
        if(s==f){
            break;
        }
    }

    int s1=0, s2=s;
    while(s1 != s2){  # Both s1 & s2 are going to same index, that means s1
                          is repeating.
        s1= nums[s1];
        s2= nums[s2];
    }
    retun s1; #
}
```