**2.1 Write code to remove duplicates from unsorted linkedList.**

public unsortedList**(**Node root**){**

HashSet**<**Integer**>** map **=** **new** HashSet**<>();**//Set para guardar los numeros una sola vez.

Node

**if(**head**!=**null**){**

map**.**add**(**head**.**val**);**//Se agrega el valor del primer nodo, no puede estar repetido

**while(**head**.**next **!=** null**){**

**if(**map**.**contains**(**head**.**next**.**val**)){**//Si el dato se repite se elimina el nodo.

head**.**next **=** head**.**next**.**next**;**

**}else{** // Si el dato no se repite se agrega a la lista y se continua iterando

map**.**add**(**head**.**next**.**val**);**

head **=** head**.**next**;**

**}**

**}**

**}**

**}**

**Complexity: 0(N).**

// This solution is for: Not Buffer Allowed

public unsortedList2**(**Node root**){**

Node first **=** root**;**

Node runner**;**

**while(**first **!=** **null){** //Make the reference node

runner **=** first**;**

**while(**runner **!=** **null){** //Iterates over all the list

**if(**runner**.**next**.**val**==**first**.**val**){**

runner**.**next **=** runner**.**next**.**next**;**

**}else{**

runner**=**runner**.**next**;**

**}**

**}**

first**=**first**.**next**;**

**}**

**}**

**Space Complexity: O(1)**

**Time Complexity: O(n2)**