

## SOLUCIÓN EVALUACIÓN CONTINUA 2

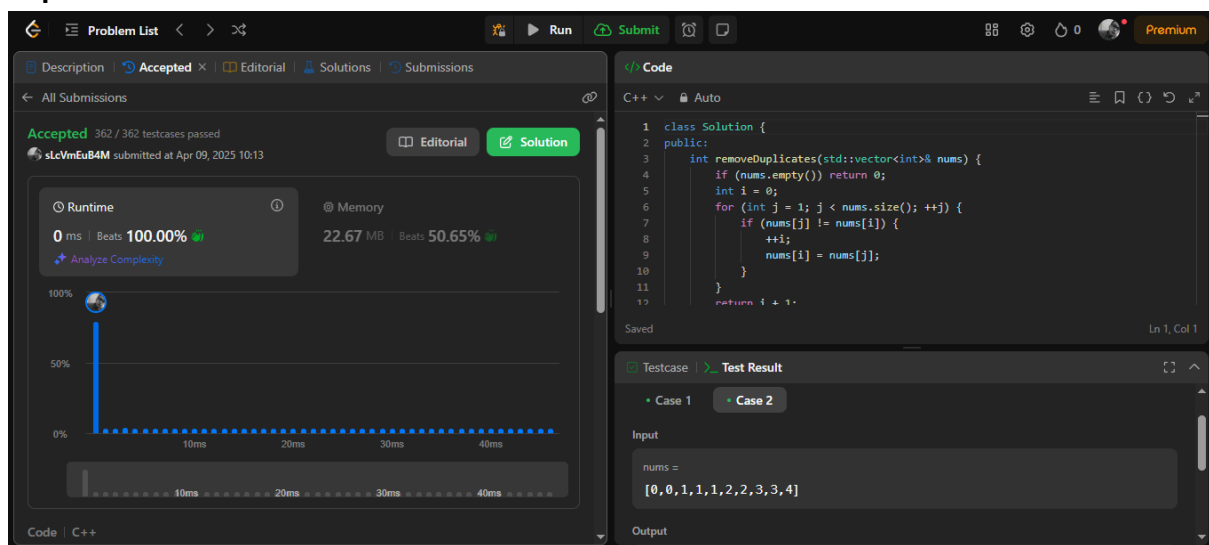
Autores: Yulinio Zavala Mariño , Vargas Iglesias Hanks, Brando Steve Lopez Carrera , Henry Rutber Quispe Sutta, Leonel Leodolfo Campuzano Diestra

### Remove Duplicates from Sorted Array.

**Alto nivel:**

1. Si el array está vacío, retornar 0.
2. Usar dos punteros: uno ("i") para seguir la posición de escritura de valores únicos, y otro ("j") para recorrer el array.
3. Comparar el valor actual ("nums[j]") con el último valor único ("nums[i]").
4. Si son distintos, avanzar "i" y copiar el valor único a "nums[i]".
5. Al final, "i + 1" representa el número de elementos únicos.

**Captura LeetCode:**

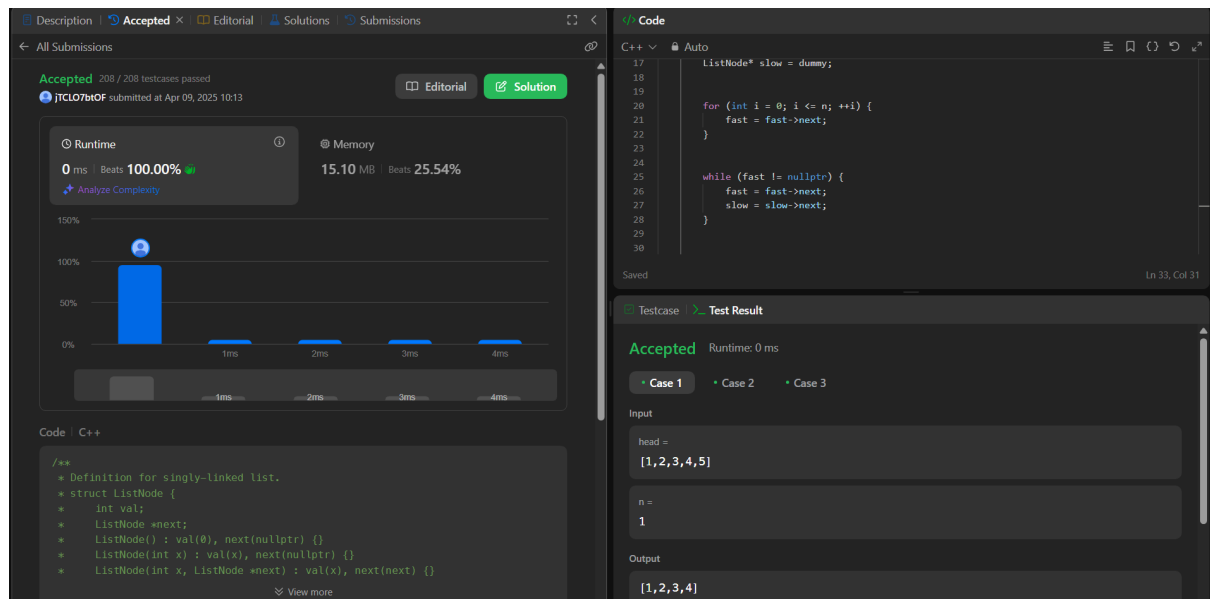


### Remove Nth Node From End of List

**Alto Nivel:**

1. Creamos un nodo auxiliar (dummy) que apunta al head.
2. Creamos dos punteros (slow, fast) que apuntan a dummy.
3. con el puntero fast, recorremos n+1 posiciones para separar una cantidad n entre slow y fast.
4. Creamos un bucle para aumentar la posición de slow y fast, hasta llegar a fast == nullptr, donde slow tomará la posición n+1 contando desde el final.
5. Almacenamos el nodo a eliminar (slow->next) en nodeToDelete
6. conectamos la posición n+1 y n-1, ya que queremos eliminar la posición n.
7. Luego eliminamos el nodo de la posición n.
8. return head (dummy->next).

**Captura LeetCode:**



## Add Two Numbers

### Alto nivel:

1. Creamos un puntero llamado sentinela con el valor de 0 y un puntero llamado current que es igual a sentinela, solo que nos servira para ir moviendonos por la lista nueva.
2. Inicializamos el valor de carry en 0 que sera el numero de las decenas de la suma.
3. Verificamos que l1 o l2 no estén vacíos o que el valor de carry sea diferente de 0
4. Inicializamos la variable sum que inicialmente tendrá el valor del carry
5. Volvemos a verificar que l1 no sea vacío para sumar los valores de cada nodo de l1 y después avanzar al siguiente nodo. Hacemos lo mismo con l2
6. Ahora que tenemos el carry, ya podremos obtener el valor del nodo resultante con  $\text{sum} \% 10$  y el carry  $\text{sum} / 10$ .
7. Creamos un nuevo nodo con el valor del nodo resultante y lo vinculamos al nodo siguiente de current. Y el carry servirá para la suma de la siguiente iteración
8. Nos movemos hacia el current siguiente y repetimos el proceso de la 2 hasta que l1 o l2 ya no tengan más dígitos por sumar.

### Captura LeetCode:

DescriptionAccepted xEditorialSolutionsSubmissions

All Submissions

Accepted 1569 / 1569 testcases passed  
jTCL07btOF submitted at Apr 09, 2025 11:00

EditorialSolution

Runtime

0 ms | Beats 100.00%

Analyze Complexity

Memory

77.30 MB | Beats 12.17%

Analyze Complexity

Runtime Range	Percentage
0-1ms	100%
1-2ms	~1%
2-3ms	~1%
3-4ms	~1%
4-5ms	~1%
5-6ms	~1%
6-7ms	~1%

Code | C++

```
/**
 * Definition for singly-linked list.
 * struct ListNode {
 *     int val;
 *     ListNode *next;
 *     ListNode() : val(0), next(nullptr) {}
 *     ListNode(int x) : val(x), next(nullptr) {}
 *     ListNode(int x, ListNode *next) : val(x), next(next) {}
 * };
 */
```

Code

C++ vAuto

```
28
29
30     carry = sum / 10;
31     int digit = sum % 10;
32
33     current->next = new ListNode(digit);
34     current = current->next;
35 }
36
37 return sentinela->next;
38 };
```

SavedLn 9, Col 6

TestcaseTest Result

AcceptedRuntime: 0 ms

Case 1Case 2Case 3

Input

I1 =  
[2,4,3]

I2 =  
[5,6,4]

Output

[7,0,8]