

Eliminar elementos duplicados consecutivos de una lista

Estructuras de Datos
Facultad de Informática - UCM

Supongamos una implementación del TAD Lista mediante listas doblemente enlazadas circulares con nodo fantasma:

```
class ListLinkedDouble {  
    // ...  
private:  
    struct Node { int value; Node *next; Node *prev; };  
    Node *head;  
    int num_elems;  
}
```

Escribe un método `dedup` que elimine los elementos duplicados de una lista, siempre que los duplicados estén en posiciones contiguas. Por ejemplo, si `xs = [1, 1, 2, 3, 3, 1, 4, 4, 4, 4]`, tras la llamada `xs.dedup()`, debe cumplirse que `xs = [1, 2, 3, 1, 4]`.

Solución

```
void ListLinkedListDouble::dedup() {
    Node *current = head->next;

    while (current->next != head) {
        if (current->value == current->next->value) {
            Node *target = current->next;
            target->next->prev = current;
            current->next = target->next;
            delete target;
            num_elems--;
        } else {
            current = current->next;
        }
    }
}
```

Coste: $\mathcal{O}(N)$, donde N es la longitud de la lista `this`, ya que el bucle hace tantas iteraciones como elementos hay inicialmente en la lista.