

Semestre 2024-1

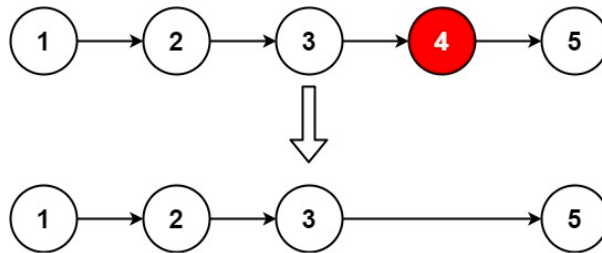
Se sugiere que cada estudiante trate de resolver los ejercicios de forma **individual** y luego los discuta en grupo.

IMPORTANTE:

- Enviar en canvas un **único archivo .cpp** (colocar las 3 soluciones en el mismo archivo .cpp).
- No se revisaran archivos **.zip** subidos a canvas.
- Colocar nombres y apellidos de los integrantes **al inicio** del archivo .cpp.
- La evaluación continua es grupal (mínimo 2 alumnos y máximo 3 alumnos).

Ejercicios

1. (6 pts) Dado el **head** de una lista simplemente enlazada, remueve el n -ésimo elemento desde el final de la lista. Luego, retorna el **head** de la lista resultante.



- Ejemplo 1:

Input: head = [1,2,3,4,5], n = 2

Output: [1,2,3,5]

- Ejemplo 2:

Output: head = [1,2], n = 1

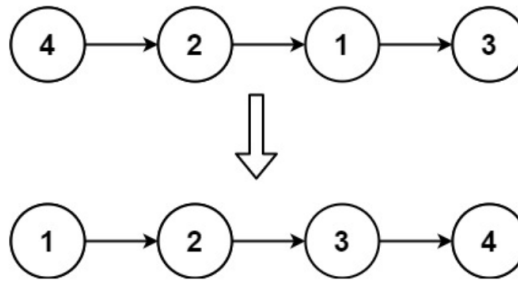
Output: [1]

2. (6 pts) Dado el **head** de una lista simplemente enlazada, ordene la lista usando **insertion sort** y retorne el **head** de la lista ordenada.

Recordemos los pasos del algoritmo de **insertion sort**:

- Iterar sobre los elementos de entrada consumiendo 1 nodo en cada repetición e insertándolo en la parte ordenada de la lista
- En cada iteración, se remueve un elemento de la lista de entrada y se inserta de forma ordenada dentro de la parte ordenada de la lista
- El algoritmo se repite siempre y cuando hayan nodos restantes por iterar

Example 1:



■ Ejemplo 1:

Input: head = [4,2,1,3]

Output: [1,2,3,4]

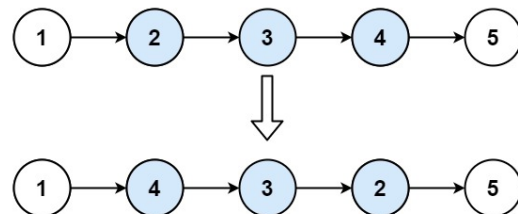
■ Ejemplo 2:

Input: head = [-1,5,3,4,0]

Output: [-1,0,3,4,5]

3. (8 pts) Dado el *head* de una lista simplemente enlazada ordenada y dos enteros *left* y *right*, donde $left \leq right$. Invierte el orden de los nodos de la lista desde la posición *left* hasta la posición *right* y retorna toda la *lista invertida*.

Example 1:



■ Ejemplo 1:

Input: head = [1,2,3,4,5], left = 2, right = 4

Output: [1,4,3,2,5]

■ Ejemplo 2:

Input: Input: head = [5], left = 1, right = 1

Output: Output: [5]