

Cuestiones ejercicio 2:

o Has implementado LinkedDeque con templates? Sea cual sea tu respuesta, justifica el motivo para tomar esta decisión.

Si. Al implementar LinkedDeque con plantillas, construimos un código genérico que puede utilizarse con cualquier tipo de datos. Con plantillas no es necesario reescribir código para utilizarlo con diferentes tipos de datos.

- o Teniendo en cuenta la implementación del TAD LinkedDeque, indica para cada una de las operaciones de la misma cuál es su coste computacional teórico. Justifica la respuesta.

Operación	Coste	Justificación
isEmpty	$O(1)$	Booleano, no hace ningún recorrido.
insertFront	$O(1)$	Crea un nuevo nodo i reasigna punteros. No hace ningún recorrido.
insertRear	$O(1)$	Crea un nuevo nodo i reasigna punteros. No hace ningún recorrido.
deleteFront	$O(1)$	Asigna front a un nuevo nodo, reasigna punteros i rear, elimina el nodo creado. No hace ningún recorrido.
deleteRear	$O(1)$	Asigna front a un nuevo nodo, reasigna punteros i rear, elimina el nodo creado. No hace ningún recorrido.
print	$O(n)$	Tantas veces como elementos tenga el TAD.
size	$O(1)$	Entero, no hace ningún recorrido.
getFront	$O(1)$	Es un getter, no hace ningún recorrido.
getRear	$O(1)$	Es un getter, no hace ningún recorrido.

- ¿Crees que la clase nodo habría sido mejor para implementarla con secuencias de simple? Justifica la respuesta.

Creo que no, usando secuencias dobles siempre sabes quién es el anterior y posterior a un elemento, y por consiguiente las implementaciones de las operaciones de la TAD se simplifican considerablemente. Por ejemplo, para hacer deleteRear no se necesita recorrer todos los elementos del TAD para reasignar los punteros, simplemente se opera con los punteros previo y siguiente de cada elemento.