

Estructuras de Datos

Tarea Colecciones Lineales y Comparadores

Los siguientes métodos deben ser incluidos en la interface `List` e implementados en las clases `ArrayList` y `LinkedList`. Cuando no se especifique, usted es libre de decidir los parámetros y tipos de dato de retorno de los métodos indicados. Cuando lo considere pertinente, incluya comentarios en sus respuestas para indicar cualquier consideración especial en cuanto a sus soluciones.

1. **`removeDuplicates(Comparator<E> comparator)`**: elimina los elementos duplicados de la lista que invoca el método. El comparador proporcionado es utilizado para determinar la igualdad de los elementos.
2. **`binarySearch(E element, Comparator<E> comparator)`**: realiza una búsqueda binaria en la lista que invoca el método.
3. **`removeElement(E element, Comparator<E> comparator)`**: elimina un elemento específico de la lista.
4. **`getIndexOf(E element, Comparator<E> comparator)`**: retorna el índice que `element` ocupa en la lista. Considere que el primer elemento de la lista tiene índice 0.
5. **`getAllIndicesOf(E element, Comparator<E> comparator)`**: retorna una colección con todos los índices en que aparece el elemento recibido en la lista.
6. **`sort(Comparator<E> comparator)`**: ordena los elementos de la lista que lo invoca utilizando el comparador proporcionado.
7. **`insertSorted(E element, Comparator<E> comparator)`**: inserta un nuevo elemento en la posición correcta para mantener el orden de la lista. El método asume que la lista que lo invoca está ordenada de manera ascendiente.²
8. **`mergeSorted(LinkedList<E> otherList, Comparator<E> comparator)`**: recibe otra lista enlazada y un comparador, y devuelve una nueva lista enlazada con los elementos de ambas listas fusionadas y ordenadas de acuerdo con el comparador proporcionado.
9. **`findUnion(LinkedList<E> otherList, Comparator<E> comparator)`**: Encuentra y retorna una nueva lista enlazada que contiene todos los elementos únicos de la lista que invoca el método y de `otherList`.
10. **`containsAll(LinkedList<E> elements, Comparator<E> comparator)`**: Verifica si todos los elementos de la lista `elements` están contenidos en la lista que invoca el método (independientemente de sus posiciones).