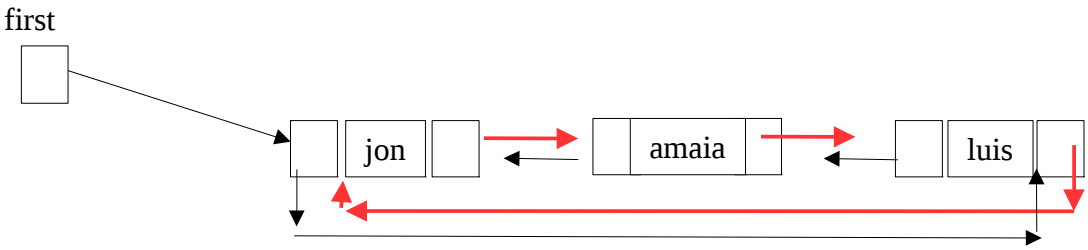
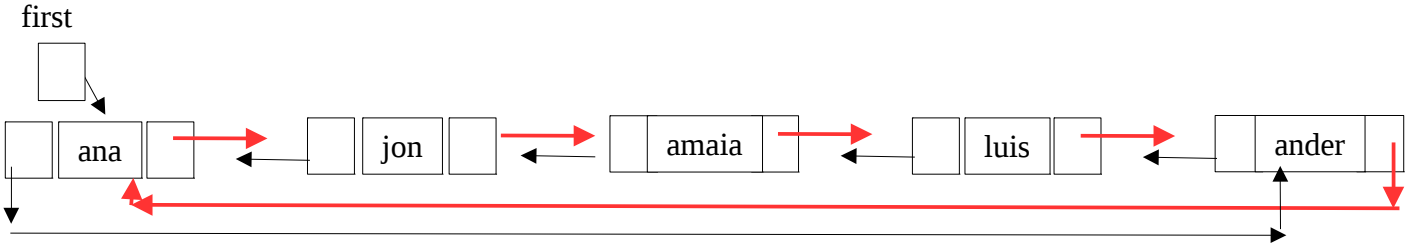
1. Borrar sublista.(1,5 puntos)
Tenemos 2 listas doblemente ligadas:

l1



l2

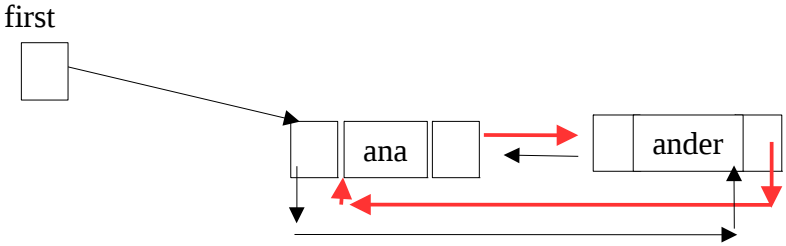


Se quiere implementar el siguiente método:

```
public class DoubleNode<T> {
   T data;
   DoubleNode<T> next;
   DoubleNode<T> prev;
}

public class DoubleLinkedList<T> {
   DoubleNode<T> first;

   public void borrarLista(DoubleLinkedList<T> subLista)
   // Precondición: "subLista" es una parte de la lista
   //     los elementos de "subLista" están en el mismo orden que en la lista principal
   // Postcondición: se han quitado los elementos de "subLista"


}
```

Por ejemplo, l2.borrarLista(l1) obtendría el siguiente resultado:

l2

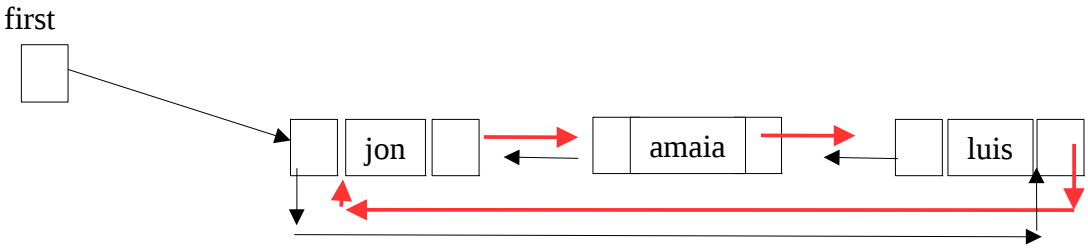


Se pide:

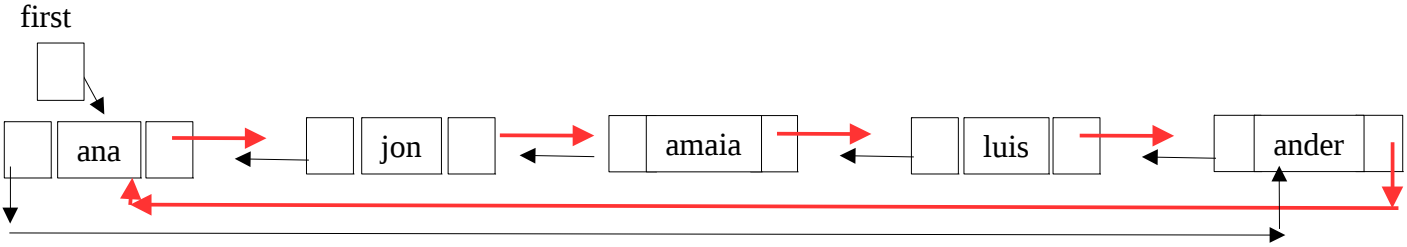
- Implementar el método
- Calcular razonadamente el coste del algoritmo

1. Ezabatu azpilista.(1,5 puntu)
Estekadura bikoitzeko bi zerrenda zirkular ditugu:

l1



l2

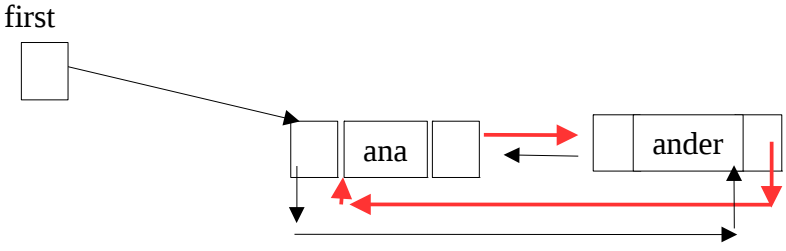


Metodo hau inplementatu nahi dugu:

```
public class DoubleNode<T> {
   T data;
   DoubleNode<T> next;
   DoubleNode<T> prev;
}

public class DoubleLinkedList<T> {
   DoubleNode<T> first;

   public void ezabatuLista(DoubleLinkedList<T> azpilista)
   // Aurrebaldintza: “azpilista” uneko listaren zati bat da
   //              “azpilista” zerrendako elementuak ordena berean daude lista nagusian
   // Postbaldintza: “azpilista” osoa kendu egin da zerrendatik


}
```

Adibidez, l2.ezabatuLista(l1) deiak zerrenda hau utziko luke:

l2



Hau eskatzen da:

- Algoritmoa inplementatu
- Kostua kalkulatu modu arrazoituan