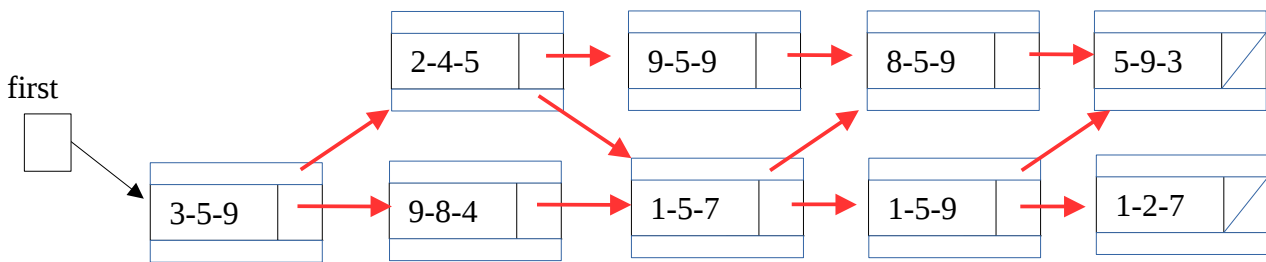


### 1. Robot (2,5 puntos)

Tenemos la siguiente lista:



La figura representa los caminos que puede seguir un robot. Se empieza siempre en el mismo punto, y a partir de él, el robot puede avanzar hacia adelante, derecha o izquierda. Por ejemplo, un posible camino a recorrer empezando desde el punto 3-5-9 sería: (izquierda, derecha, adelante, izquierda). La lista de puntos recorridos (cada punto es representado por sus coordenadas) será: (3-5-9, 2-4-5, 1-5-7, 1-5-9, 5-9-3)

Se quiere implementar la función *obtCoordenadas* que, dada una lista con los movimientos a realizar por el robot, nos devuelva la lista de los puntos recorridos:

```
public class Node {
    String coord;
    Node next;
    Node left;
    Node right;
}
public class Camino {
    Node first;

    public CircularLinkedList<String> obtCoordenadas(ArrayList<String> acciones)
        // pre: la lista acciones tiene al menos un elemento, e indica los
        //       movimientos a realizar por el robot
        //       El recorrido no produce error (nunca se intenta avanzar por un
        //       enlace con valor null)
        // post: devuelve una nueva lista, que contendrá
        //       las coordenadas de los puntos recorridos
        //       La lista original no se ha modificado
    }

    public class CircularLinkedList<T> {
        // lista circular con enlace al último elemento
        NodeCircularLinkedList<T> last;
    }
    public class NodeCircularLinkedList<T> {
        T data;
        NodeCircularLinkedList<T> next;
    }
}
```

Por ejemplo, la llamada a *obtCoordenadas*((izquierda, derecha, adelante, izquierda)) devolverá una lista que contendrá a (3-5-9, 2-4-5, 1-5-7, 1-5-9, 5-9-3).

Se pide:

- Implementar el algoritmo
- Establecer de manera razonada el coste del algoritmo.

**Nota:** todos los métodos de las clases **Camino** y **CircularLinkedList** que se utilicen deberán ser implementados.