Reordena llista X48487_ca

Implementeu el mètode

```
void reordena ()
```

de la classe `List`, la implementació de la qual us donem al fitxer `list.hpp`.
Després de cridar el mètode `L.reordena()`, on `L` és una llista de tipus `List<T>` **doblement enllaçada**, la llista `L` (el paràmetre implícit) està ordenada creixentment.
Ara bé, cal tenir en compte que abans de fer la crida (com a precondició del mètode), `L` té **un sol** element desordenat. Una cosa important és que aquest element desordenat *cal moure'l cap al final de la llista* per tal que `L` estigui completament ordenada. No hi haurà cap element desordenat tal que, per tal d'ordenar la llista, calgui moure'l cap al principi de la llista.
Per tant, el mètode `reordena` recol·locarà l'element no ordenat al lloc que li correspon per tal que la llista quedi ordenada.
Per exemple, si tenim la llista:

```
L = [1 3 11 7 9 15]
```

Després de fer la crida a `L.reordena()` tindrem:

```
L = [1 3 7 9 11 15]
```

Tingueu en compte que **no podeu assumir res** sobre la mida de la llista.
La puntuació que podeu obtenir és la següent:

1. Solució correcta en els jocs de proves públics: 5 punts.
2. Solució correcta en els jocs de proves públics, especificació de la funció, invariant i funció fita: 8 punts.
3. Solució correcta en els jocs de proves públics i privats: 7 punts.
4. Solució correcta en els jocs de proves públics i privats, especificació de la funció, invariant i funció fita: 10 punts.

Quan diem *especificació de la funció, invariant i funció fita* volem dir que hi ha de ser **tot**. Dit altrament: no es donarà una fracció dels 3 punts si doneu només, per exemple, l'especificació de la funció, o només l'invariant i la fita. Se us donarà la bonificació dels 3 punts únicament si feu totes 3 coses **correctament**.

Entrada

El mètode rep una instància de tipus `List<T>` amb un element desordenat.

Sortida

El mètode reordena la llista del paràmetre implícit.

Observació

Heu d'enviar la solució comprimida en un fitxer .tar:
`tar cvf program.tar list.cpp`
Observeu que per compilar us donem el `Makefile`, la capçalera del mòdul funcional `list.hpp`, que conté tota la implementació de tots els mètodes, (llevat del que heu d'implementar vosaltres al programa `list.cpp`) i el programa principal `program.cpp`.
La solució es compilarà perquè el fitxer `list.hpp` (que no heu de modificar) té la següent línia: `include "list.cpp"`.
Per tant, no cal que modifiqueu cap fitxer, simplement cal que envieu el fitxer `list.cpp` comprimit (tal i com hem explicat) amb aquesta capçalera:
`void reordena ()`

Exemple d'entrada 1

```
6
1 3 11 7 9 15
```

Exemple de sortida 1

```
[6] 1 3 7 9 11 15
```

Exemple d'entrada 2

```
4
2 5 3 4
```

Exemple de sortida 2

```
[4] 2 3 4 5
```

Informació del problema

Autor : PRO1-Vilanova
Generació : 2024-01-26 16:02:48