
Comprimir llista dispersa recursiva

W94440_ca

Una llista d'enters és **dispersa** si la major part dels elements són 0.
Per exemple, la llista $l = [0, 0, 0, 3, 0, 0, 0, 1]$ és dispersa.

Una llista dispersa es pot comprimir de manera que només emmagatzemi parells ($\langle \text{posicio}, \text{valor} \rangle$) d'aquells elements no nuls. Cal tenir present que el primer parell que emmagatzema conté el nombre d'elements de la llista original.

Donada la llista anterior, la versió comprimida seria: $\text{comprimir}(l) = [\langle 8, 0 \rangle, \langle 3, 3 \rangle, \langle 7, 1 \rangle]$

Implementa una acció **RECURSIVA** que donada un llista d'enters torna la versió comprimida d'aquesta llista.

Interfície

La capçalera de la funció és la següent:

```
// Pre: cert
// Post: torna una llista que conté la llista "l" comprimida.
void comprimir(const list<int> &l, list<pair<int,int>> &compr)
```

Observació

Les funcions i accions que creïs han de treballar només amb llistes (la classe `list` de la biblioteca STL). Has de trobar una solució **RECURSIVA** i eficient del problema. En particular, no hi hauria d'haver cap bucle en cap de les funcions/accions que implementis. Si crees funcions/accions auxiliars, afegeix-hi les corresponents **Precondició** (Pre) i **Postcondició** (Post). En les crides recursives inclou la **hipòtesi d'inducció** (HI) i la **funció de fita** (FF).

IMPORTANT: Només cal enviar el procediment demanat; el programa principal serà ignorat.

Informació del problema

Autoria: Bernardino Casas

Generació: 2026-01-25T13:21:24.492Z

© Jutge.org, 2006–2026.
<https://jutge.org>