

P2: Metodologies Formals

CRISTINA IZQUIERDO

CRISTÒFOL DAUDÉN

ALEIX MARINÉ

TAD llista ordenada

Especificació del TAD llista:

Insereix: a la llista 'll' afegeix l'element 'e'

EliminaP: elimina l'element de la llista 'll' que està a la posició 'p'

EliminaV: elimina l'element de la llista 'll' que té per valor 'e'

Buida: buida la llista 'll'

Mida: retorna la mida de la llista 'll'

Suma: retorna la suma dels elements de la llista 'll'

Positiva: converteix a positius tots els elements de la llista 'll'

HiEs: retorna cert si dins la llista 'll' hi ha l'element 'e'

Fusiona: emplena una llista 'lld' amb els elements de les llistes 'll1' i 'll2'

TAD llista ordenada

Max: Retorna l'element màxima de la llista

Min: Retorna l'element mínim de la llista

Inserció: Ordena el vector utilitzant l'algorisme d'inserció

OmplirRandom: Omple el vector amb valors aleatoris positius ordenats

OmplirRandomMig: Omple el vector fins a la meitat utilitzant valors aleatoris negatius

Definicions Formals dels mètodes

/**

* Resum: retorna la posició de l'element "var" dins la llista 'll' si hi és. Sinó, retorna -1

* Precondició: la llista ll està ordenada

* Postcondició: $\text{cent} = (\text{existeix un } \alpha : ll.v[\alpha] == \text{cent}) \mid -1$

* Errors: no es defineixen

* Paràmetres:

* - ll, var

*/

int HiEs(t_llista ll, int var){ //Algorisme de cerca dicotòmica per buscar un element

/**

* Resum: reordena els elements de la llista 'll' utilitzant

* l'algorisme d'inserció

* Precondició: cert

* Postcondició: els elements de la llista estan ordenats

* Errors: no es defineixen

* Paràmetres:

* - ll és del tipus t_llista

*/

void insercio(t_llista *ll)

/**

* Resum: Afegeix a la llista 'll' l'element 'e'

* Precondició: la llista no està plena

* Postcondició: s'afegeix l'element 'e' a la llista 'll' i

* no es modifica si 'll' és buida o bé 'e' no s'ha trobat

* Errors: la llista està plena

* Paràmetres:

* - ll és del tipus t_llista

* - e és del tipus enter

*/

void insereix(t_llista *ll, int e){

/**

* Resum: Elimina l'element de la llista 'll' de la posició 'p'

* Precondició: $p \geq ll.n$

* Postcondició: 'll' perd l'element de la posició 'p' i

* no es modifica si 'll' és buida o bé 'p' no s'ha trobat

* Errors: $p \geq ll.n$

* Paràmetres:

* - ll és del tipus t_llista

* - p és del tipus natural

*/

void eliminaP (t_llista *ll, unsigned int p){

/**

* Resum: Elimina el valor de la llista 'll' que té per valor 'e'

* Precondició: existeix un alfa menor que n i major que 0 tal que v[alfa] = e

* Postcondició: 'll' perd tots els elements que tenen per valor 'e' i

* no es modifica si 'll' és buida o bé 'e' no s'ha trobat

* Errors: l'element e no existeix dins del vector

* Paràmetres:

* - ll és del tipus t_llista

* - e és del tipus enter

*/

void eliminaV (t_llista *ll, int e){

/**

* Resum: Retorna el número d'elements de la llista 'll'

* Precondició: cert

* Postcondició: cert

* Errors: no es defineixen

* Paràmetres:

* - ll és del tipus t_llista

*/

int mida (t_llista ll){

```
/**
```

```
* Resum: Imprimeix per pantalla la llista 'll'
```

```
* Precondició: cert
```

```
* Postcondició: cert
```

```
* Errors: no es defineixen
```

```
* Paràmetres:
```

```
* - ll és del tipus t_llista
```

```
*/
```

```
void imprimeix(t_llista *ll){
```

/**

* Resum: Omple la llista 'll' amb elements aleatoris, fins a la meitat de la seva capacitat.

* Precondició: cert

* Postcondició: La llista 'll' és mig plena i ordenada

* Errors: no es defineixen

* Paràmetres:

* - ll és del tipus t_llista

*/

void omplirRandomMig(t_llista *ll){

/**

* Resum: retorna la suma dels elements de la llista 'll'

* Precondició: cert

* Postcondició: suma de tots els elements de 'll'

* Errors: si la suma de tots els elements supera MAX_INT el programa pot comportar-se de forma inconsistent

* Paràmetres:

* - ll

*/

int suma(t_llista ll){

```
/**
```

```
* Resum: converteix a positius tots els elements de la llista 'll'
```

```
* Precondició: cert
```

```
* Postcondició: per qualsevol alfa tal que alfa mes gran que 0 i menor que n_elem llavors v[alfa]  
>= 0 & taula ordenada
```

```
* Errors: la llista està buida
```

```
* Paràmetres:
```

```
* - ll
```

```
*/
```

```
void positiva(t_llista *ll){
```

/**

* Resum: Fusiona dues llistes sobre una llista destí

* Precondició: la suma del nombre d'elements de les dues llistes origen supera N_MAX

* Postcondició: retorna cert i 'lld' conté els elements de les llistes origen, mantenint l'ordre o bé si $ll1.n + ll2.n > N_MAX$ reetorna fals

* Errors: la suma del nombre d'elements de les dues llistes origen supera N_MAX

* Paràmetres:

* - lld, ll1, ll2 són del tipus llista

*/

bool fusiona(t_llista ll1, t_llista ll2, t_llista *lld){

/**

* Resum: buida la llista ll

* Precondició: cert

* Postcondició: numero d'elements de la llista es 0

* Errors: no es defineixen

* Paràmetres:

* - ll es la llista t

*/

void buida (t_llista *ll){

```
/**
```

```
* Resum: Omple la llista 'll' amb elements aleatoris.
```

```
* Precondició: cert
```

```
* Postcondició: La llista 'll' és plena i ordenada
```

```
* Errors: no es defineixen
```

```
* Paràmetres:
```

```
* - ll és del tipus t_llista
```

```
*/
```

```
void omplirRandom(t_llista *ll){
```



```
/**
```

```
* Resum: Retorna l'element mínim de la llista
```

```
* Precondició: cert
```

```
* Postcondició: l'element retornat es el primer de la llista
```

```
* Errors: no es defineixen
```

```
* Paràmetres:
```

```
* - ll es la llista t
```

```
*/
```

```
int min(t_llista t){
```

/**

* Resum: Retorna l'element màxim de la llista

* Precondició: cert

* Postcondició: l'element retornat es l'últim de la llista

* Errors: no es defineixen

* Paràmetres:

* - ll es la llista t

*/

int max(t_llista t){

/**

* Resum: Menú per a seleccionar la funció desitjada

* Precondició: cert

* Postcondició: no es modifica res

* Errors: no es defineixen

*/

int opcio(){

Asserts

S'han col·locat els asserts en punts estratègics dins dels mètodes. Per exemple:

- EliminaP rep una posició de fora de l'array
- EliminaV rep un valor que no es troba dins de l'array.
- Al main, una introducció d'un int és incorrecta (en la funcio atoi())
- A insereix la llista esta plena

S'ha creat un interruptor global d'asserts en la variable bool activa_asserts per a depurar el codi més fàcilment.