

(format MD)

### # Ex3

Igual que a l'exercici anterior cal destacar que s'ha canviat la firma de la funció crear ja que sino resulta impossible implementar la funció Destruir() ja que es impossible recuperar l'adreça original creada per la funció malloc al reservar memoria

#### ### Tasques realitzades:

- \* S'ha complert l'especificació de la doble llista encadenada
- \* S'ha fet un joc de proves amb aquesta llista
- \* S'ha valorat el cost i s'ha fet el gràfic corresponent demanat a l'enunciat
- \* S'ha intentat implementar una skip list. El codi s'ha escrit pero hi han errors d'execució

#### #### Detalls d'implementació

##### ##### Llista doblement encadenada

S'ha decidit que els elements s'insereixen per darrere del PDI, de tal forma que no es poden afegir per davant, degut a això:

S'ha decidit que quan el PDI apunta a NULL es considera el principi de la llista i llavors els elements es fiquen a la primera posició de la llista.

Quan s'elimina l'element apuntat per PDI es va a parar a l'element del darrere, excepte si aquest es l'últim, tornant al penúltim.

##### ##### Skip list

Es van seguir les instruccions donades a la practica però hi van haver errors a la rutina inserir i vaig passar la major part del temps intentant trobar l'error.

##### ##### Comandes per a correr l'algorisme

- Testing de la llista doblement encadenada: ``make runlisttest``
- Testing de la skiplist (compte! conté errors): ``make runskiplisttest``

