Algoritmos y Estructuras de Datos.
RTPL1. Recup Trábajo Práctico de Laboratorio 1. [2014-09-23]

Ejercicios

[Ej. 1] **[expand]** Escribir una función `void expand(list<int>  &L,int m);` que transforma los elementos de una lista `L` de tal forma que todos los elementos de `L` resulten ser menores o igual que `m`, pero de tal forma que su suma se mantenga inalterada. por ejemplo si `m=3` podemos dividir a `10` en `3,3,3,1`. Es decir si `L=(7,2,3,1,4,5)`, entonces despues de hacer `expand(L,2)` debe quedar `L=(2,2,2,1,2,2,1,1,2,2,2,2,1)`.

[Ej. 2] **[ascendente]** Escribir una función `int ascendente(list <int> &L, list<list<int> > &LL)` que, dada una lista `L`, genera una lista de listas `LL` de tal forma de que cada sublista es ascendente. Por ejemplo, si la lista es `L=(0,5,6,9,4,3,9,6,5,5,2,3,7)`, entonces hay 6 corridas ascendentes, a saber: `(0,5,6,9)`, `(4)`, `(3,9)`, `(6)`, `(5,5)` y `(2,3,7)`. Por lo tanto la función debe retornar `LL=((0,5,6,9,4,3,9,6,5,5,2,3,7), (0,5,6,9), (4), (3,9), (5,5), (2,3,7))`.

[Ej. 3] **[deja1solo]** Escribir una función `void deja1solo(list<int> &L);` de tal forma que. si vamos dividiendo a `L` en rangos de pares e impares consecutivos, deja el primero de cada rango, por ejemplo si `L=(5 4 5 3 4 9 10 4 9 3 2 10 3 7 5 3)` entonces debe dejar `L=(5 4 5 4 9 10 4 9 3 2 3)` Para `L=(10 10 1 9 7 4 4 10 7 1 1 4 10 1 1 7)` debe dejar `L=(10 1 4 7 4 1)`. El algoritmo debe ser in-place, sólo debe borrar elementos.

mstorti@galileo/aed-2.0.5-67-gcc0cf57/Tue Sep 23 12:39:33 2014 -0300