

Lenguaje Natural

Tenemos el árbol TArbolActores que tiene un árbol con los actores y sus películas. En el método de este árbol creamos una lista de co-actores. Luego buscamos el actor en la lista de actores. Si el actor existe entonces cargamos la lista de películas de ese actor en una lista. Si el actor tiene películas (existen) recorremos la lista de películas y para cada una verificamos los actores en ella agregándolos a una lista que retorna.

Precondiciones

- La lista de actores no puede estar vacía.

Postcondiciones

- Los árboles originales no deben cambiar.
- Las listas originales no cambian.
- El algoritmo retorna una lista con elementos o vacía.

En TarbolActores. coactores (Actor de tipo (arbol))
: devuelve una lista de tipo Actor.

Inicio

Lista coactores de tipo Actor \leftarrow nueva lista

Elemento elActor de tipo Actor \leftarrow arbol. buscar (Actor
// $O(\log(n))$)

Si (elActor \neq nulo) Entonces

Lista Films de tipo Film \leftarrow elActor. obtenerDatos (

Si (Films \neq nulo) Entonces obtener Films ()
// $O(n)$

Nodo pelicula de tipo Film \leftarrow Films. obtenerPrimer ()
// $O(1)$

Mientras (pelicula \neq nulo) HACER

PeliculaAux de tipo Film \leftarrow pelicula. obtenerPrimer
// $O(1)$

Lista actores de tipo Actor \leftarrow PeliculaAux.

Si (actores \neq nulo) Entonces obtener Actores ()
// $O(n)$

Nodo coact de tipo Actor \leftarrow actores.

Mientras (coact \neq nulo) HACER obtenerPrimer ()
// $O(1)$

Si (coact. etiqueta comparado (elActor) \neq 0
y coact. buscar (coact. etiqueta) \neq nulo
Entonces coactores. insertar (coact)

coact \leftarrow coact. getSiguinte ()

Fin Mientras

Fin Si

Pelicula \leftarrow pelicula. getSiguinte

Problema Stable - Pasa 3

(2)

! ! Fin Mientras

! Fin Si

Fin Si

Retornar coactores (la lista resultado)

FIN

Orden de tiempo euclídeo.

Cantidad de actores x cantidad de coactores y las películas.