



DEPARTAMENTO  
DE COMPUTACION

Facultad de Ciencias Exactas y Naturales - UBA

# Recuperatorio

## Ejercicio X1

27 de julio de 2020

Algoritmos y Estructuras de Datos II

Integrante	LU	Correo electrónico
Rodriguez, Miguel	57/19	mmiguerodriguez@gmail.com



**Facultad de Ciencias Exactas y Naturales**  
Universidad de Buenos Aires

Ciudad Universitaria - (Pabellón I/Planta Baja)

Intendente Güiraldes 2610 - C1428EGA

Ciudad Autónoma de Buenos Aires - Rep. Argentina

Tel/Fax: (54 11) 4576-3300

<https://exactas.uba.ar>

## Ejercicio

TAD ADH( $\alpha$ )

**géneros**       $\text{adh}(\alpha)$

**igualdad observacional**

$$(\forall a, a' : \text{adh}) \left( a =_{\text{obs}} a' \iff \left( \text{raíz}(a) =_{\text{obs}} \text{raíz}(a') \wedge \text{hijos}(a) =_{\text{obs}} \text{hijos}(a') \right) \right)$$

**observadores básicos**

$\text{raíz} : \text{adh}(\alpha) \longrightarrow \alpha$

$\text{hijos} : \text{adh}(\alpha) \longrightarrow \text{dicc}(\text{nat}, \text{adh}(\alpha))$

**generadores**

$\text{cons} : \alpha \times \text{dicc}(\text{nat}, \text{adh}(\alpha)) \longrightarrow \text{adh}(\alpha)$

**otras operaciones**

$\text{altura} : \text{adh}(\alpha) \longrightarrow \text{nat}$

$\text{hojas} : \text{adh}(\alpha) \longrightarrow \text{conj}(\alpha)$

$\text{ramas} : \text{adh}(\alpha) \longrightarrow \text{conj}(\text{secu}(\alpha))$

$\text{maxAlturaHijos} : \text{dicc}(\text{nat}, \text{adh}(\alpha)) \times \text{conj}(\text{nat}) \longrightarrow \text{nat}$

$\text{hojasHijos} : \text{dicc}(\text{nat}, \text{adh}(\alpha)) \times \text{conj}(\text{nat}) \longrightarrow \text{conj}(\alpha)$

$\text{ramasHijos} : \text{dicc}(\text{nat}, \text{adh}(\alpha)) \times \text{conj}(\text{nat}) \longrightarrow \text{conj}(\text{secu}(\alpha))$

$\text{prefijarEnTodos} : \alpha \times \text{conj}(\text{secu}(\alpha)) \longrightarrow \text{conj}(\text{secu}(\alpha))$

**axiomas**

$\text{raíz}(\text{cons}(r, h)) \equiv r$

$\text{hijos}(\text{cons}(r, h)) \equiv h$

$\text{altura}(a) \equiv \text{if vacío?}(\text{claves}(\text{hijos}(a))) \text{ then}$

1

else

1 + maxAlturaHijos(hijos(a), claves(hijos(a)))

fi

$\text{maxAlturaHijos}(as, keys) \equiv \text{if } \#(keys) = 1 \text{ then}$

altura(obtener(dameUno(keys), as))

else

max(altura(obtener(dameUno(keys), as)),

maxAlturaHijos(as, sinUno(keys)))

fi

$\text{hojas}(a) \equiv \text{if vacío?}(\text{claves}(\text{hijos}(a))) \text{ then}$

{ raíz(a) }

else

hojasHijos(hijos(a), claves(hijos(a)))

fi

```

hojasHijos(as, keys)  $\equiv$  if  $\#(keys) = 0$  then
     $\emptyset$ 
else
    hojas(obtener(dameUno(keys), as))  $\cup$  hojasHijos(as,
    sinUno(keys))
fi

ramas(a)  $\equiv$  if vacío?(claves(hijos(a))) then
    {< raíz(a) >}
else
    prefijarEnTodos(raíz(a), ramasHijos(hijos(a), claves(hijos(a))))
fi

ramasHijos(as, keys)  $\equiv$  if  $\#(keys) = 0$  then
     $\emptyset$ 
else
    ramas(obtener(dameUno(keys), as))  $\cup$  ramasHijos(as,
    sinUno(keys))
fi

prefijarEnTodos(e, cs)  $\equiv$  if vacío?(cs) then
     $\emptyset$ 
else
    Ag(e • dameUno(cs), prefijarEnTodos(e, sinUno(cs)))
fi

```

**Fin TAD**

## Justificaciones

**Generales:** La idea y estructura en la mayoría de las funciones es muy parecida. Además de pasar recursivamente los hijos de cada ADH para poder calcular lo que se nos pide, agregamos como parámetro las claves de este diccionario para así poder recorrerlos. Luego, en cada paso, vamos sacando las claves hasta quedarnos sin.

**Altura:** Por el ejemplo dado en el enunciado, me da a entender que un ADH sin hijos tiene altura = 1. En caso de tener hijos, usamos MAXALTURAHIJOS que a su vez llama a ALTURA en caso de haber sólo una clave (para terminar la recursión), y en el caso de que haya más de 1 clave, se llama recursivamente. Podemos notar que MAXALTURAHIJOS obtiene el máximo de todos los resultados recursivos.

**Hojas:** La idea es la misma que para ALTURA. En caso de que el ADH no tenga hijos, devolvemos su raíz ya que va a ocurrir que es una hoja. En el caso contrario, buscamos las hojas de sus hijos que, en el final de la recursión, va a hacer una unión de todas las raíces de los ADH que no tengan hijos.

**Ramas:** Para esta funcionalidad, continuamos con la misma idea de pasar las claves del diccionario de los hijos para poder recorrerlos. En el caso de que un ADH no tenga hijos, entonces el resultado del conjunto de secuencias de sus ramas va a ser únicamente la raíz. En el caso contrario, lo que hacemos es calcular las ramas de los hijos y además prefijar en todas la raíz.