



DEPARTAMENTO
DE COMPUTACION

Facultad de Ciencias Exactas y Naturales - UBA

Trabajo Práctico 2: Diseño

Primer cuatrimestre - 2016

Algoritmos y Estructuras de Datos II

Grupo 22

Integrante	LU	Correo electrónico
BENZO, Mariano	198/14	marianobenzo@gmail.com
FARIAS, Mauro	821/13	farias.mauro@hotmail.com
GUTTMAN, Martin	686/14	mdg_92@yahoo.com.ar

Instancia	Docente	Nota
Primera entrega		
Segunda entrega		



Facultad de Ciencias Exactas y Naturales
Universidad de Buenos Aires

Ciudad Universitaria – Pabellón I (Planta Baja)

Intendente Güiraldes 2160 – C1428EGA

Ciudad Autónoma de Buenos Aires – Rep. Argentina

Tel/Fax: (+54 +11) 4576-3300

<http://www.exactas.uba.ar>

Índice

1. Registro	2
1.1. Interfaz	2
1.2. Representación	4
1.3. Algoritmos	5

1 Registro

1.1 Interfaz

se explica con REGISTRO

usa nat, dato, campo, conjTrie, string, itConj

géneros reg, itReg

Operaciones

$\text{NREG}() \longrightarrow res : \text{reg}$

Pre $\equiv \{\text{true}\}$

Post $\equiv \{res =_{\text{obs}} \emptyset\}$

Descripción: Crea un registro nuevo, vacío

Complejidad: $O(1)$

Aliasing: No tiene Aliasing

$\text{CAMPOS}(\text{in } r : \text{reg}) \longrightarrow res : \text{itConj}(\text{campo})$

Pre $\equiv \{\text{true}\}$

Post $\equiv \{res =_{\text{obs}} \text{alias}(res, \text{claves}(r))\}$

Descripción: Devuelve un conjunto de campos que son claves del registro ingresado por parametro

Complejidad: $O(1)$

Aliasing: Se devuelve un iterador al conjunto, hay aliasing

$\text{BORRAR?}(\text{in } crit : \text{reg}, \text{ in } r : \text{reg}) \longrightarrow res : \text{bool}$

Pre $\equiv \{\#campos(crit) = 1\}$

Post $\equiv \{res =_{\text{obs}} \text{borrar?}(crit, r)\}$

Descripción: Devuelve true si y solo si todos los campos de crit pertenecen a campos de r.

Complejidad: $O(1)$

$\text{AGCAMPOS}(\text{in/out } r_1 : \text{reg}, \text{ in } r_2 : \text{reg})$

Pre $\equiv \{r_1 =_{\text{obs}} r_{10}\}$

Post $\equiv \{r_1 =_{\text{obs}} \text{agregarCampos}(r_{10}, r_2)\}$

Descripción: agrega los datos de los campos faltantes de r2 a los campos de r1

Complejidad: $O(1)$

$\text{COPIARCAMPOS}(\text{in/out } r_1 : \text{reg}, \text{ in } cc : \text{conj}(\text{campo}), \text{ in } r_2 : \text{reg})$

Pre $\equiv \{r_1 =_{\text{obs}} r_{10} \wedge cc \in \text{campos}(r_2)\}$

Post $\equiv \{r_1 =_{\text{obs}} \text{copiarCampos}(cc, r_{10}, r_2)\}$

Descripción: copia los datos que se encuentran en los campos de r2 a r1

Complejidad: $O(1)$

$\text{COINCIDEALGUNO}(\text{in } r_1 : \text{reg}, \text{ in } cc : \text{conj}(\text{campo}), \text{ in } r_2 : \text{reg}) \longrightarrow res : \text{bool}$

Pre $\equiv \{cc \subseteq \text{campos}(r_1) \cap \text{campos}(r_2)\}$

Post $\equiv \{res =_{\text{obs}} \text{coincideAlguno}(r_1, cc, r_2)\}$

Descripción: Devuelve true si y solo si alguno de los campos(dato) de cc pertenece a r1 y r2

Complejidad: $O(1)$

$\text{COINCIDENTODOS}(\text{in } r_1 : \text{reg}, \text{ in } cc : \text{conj}(\text{campo}), \text{ in } r_2 : \text{reg}) \longrightarrow res : \text{bool}$

Pre $\equiv \{cc \subseteq \text{campos}(r_1) \cap \text{campos}(r_2)\}$

Post $\equiv \{res =_{\text{obs}} \text{coincidenTodos}(r_1, cc, r_2)\}$

Descripción: Devuelve true si y solo si todos los campos(dato) de cc pertenecen a r1 y r2

Complejidad: $O(1)$

ENTODOS(**in** $c : \text{campo}$, $\text{in } cr : \text{conj}(\text{registro})$) $\longrightarrow res : \text{bool}$

Pre $\equiv \{\text{true}\}$

Post $\equiv \{res =_{\text{obs}} \text{enTodos}(c, cr)\}$

Descripción: Devuelve true si y solo si campo c pertenece a los campos de cada uno de los registros cr

Complejidad: $O(\#(cr))$

UNIRREGISTROS(**in** $c : \text{campo}$, $\text{in } r_1 : \text{reg}$, $\text{in } r_2 : \text{reg}$) $\longrightarrow res : \text{registro}$

Pre $\equiv \{ccampos(r_1) \wedge c(campos(r_2))\}$

Post $\equiv \{res =_{\text{obs}} \text{combinarTodos}(c, r_1, ag(\emptyset, r_2))\}$

Descripción: Devuelve el registro que combina los valores de r_1 y r_2

Complejidad: $O(1)$

Aliasing: r_1 y r_2 son tomados por referencia

1.2 Representación

se representa con `DiccString(campo, dato)`

Invariante de representación

$\text{Rep} : \widehat{\text{Dicc}} \longrightarrow \text{boolean}$

$(\forall d : \widehat{\text{Dicc}})$
 $\text{Rep}(d) \equiv \text{true}$

Función de abstracción

$\text{Abs} : \widehat{\text{DiccTrie}}\ d \longrightarrow \widehat{\text{reg}}$

$\{\text{Rep}(d)\}$

$(\forall d : \widehat{\text{DiccTrie}})$

$\text{Abs}(d) \equiv r : \widehat{\text{reg}} \mid \text{if } \text{vacía?}(d.\text{claves}) \text{ then}$

vacío

else

$\text{definir}(\text{prim}(d).\text{claves}, \text{prim}(d).\text{significado}, \text{Abs}(\text{fin}(d)))$

fi

1.3 Algoritmos

campos (in $r : \text{reg}$) $\rightarrow res : \text{Conj}(\text{campo})$	
$cn \leftarrow \text{vacío}();$	$O(1)$
$it \leftarrow \text{crearItDicc}(r);$	
while haySiguiente(it) do;	$O(1)$
agregar(cn , $\text{clave.siguiente}(it)$);	$O(1)$
avanzar(it);	$O(1)$
end while	
$res \leftarrow cn;$	$O(1)$
<hr/>	
$O(1)$	
borrar? (in $crit : \text{reg}$ in $r : \text{reg}$) $\rightarrow res : \text{bool}$	
$cc \leftarrow \text{crit.campos};$	$O(1)$
$res \leftarrow \text{coincidenTodos}(crit, cc, r);$	$O(1)$
<hr/>	
$O(1)$	
agregarCampos (in $r_1 : \text{reg}$ in $r_2 : \text{reg}$) $\rightarrow res :$	
$cc \leftarrow r_2.\text{campos} / r_1.\text{campos};$	$O(1)$
$res \leftarrow \text{copiarCampos}(cc, r_2, r_1);$	$O()$
<hr/>	
$O(1)$	
copiarCampos (in $cc : \text{conj}(\text{campo})$ in/out $r_1 : \text{reg}$, in $r_2 : \text{reg}$)	
$itcc \leftarrow \text{crearItConj}(cc);$	$O(1)$
while haySiguiente(itcc) do;	$O(1)$
definirRapido($\text{siguiente}(itcc)$, $\text{obtener}(\text{siguiente}(itcc) r_2)$, r_1);	$O(1)$
avanzar(itcc);	$O(1)$
end while	
<hr/>	
$O(1)$	
coincideAlguno (in $r_1 : \text{reg}$ in $cc : \text{conj}(\text{campo})$ in $r_2 : \text{reg}$) $\rightarrow res : \text{bool}$	
$res \leftarrow \text{false};$	$O(1)$
if $\neg \text{vacío?}(cc)$ then;	$O(1)$
$itcc \leftarrow \text{crearItConj}(cc);$	$O(1)$
while haySiguiente(itcc) do; $O(1)$	
if $\text{obtener}(\text{siguiente}(itcc), r_1) == \text{obtener}(\text{siguiente}(itcc), r_2)$ then;	
$res \leftarrow \text{true};$	$O(1)$
end if	
avanzar(itcc);	$O(1)$
end while	
<hr/>	
$O(1)$	
coincidenTodos (in $r_1 : \text{reg}$ in $cc : \text{conj}(\text{campo})$ in $r_2 : \text{reg}$) $\rightarrow res : \text{bool}$	
$res \leftarrow \text{true};$	$O(1)$
$itcc \leftarrow \text{crearItConj}(cc);$	$O(1)$
while haySiguiente(itcc) $\wedge res$ do; $O(1)$	
if $(\text{obtener}(\text{siguiente}(itcc) r_1) == \text{obtener}(\text{siguiente}(itcc) r_2))$ then;	
$res \leftarrow \text{false};$	$O(1)$
end if	
avanzar(itcc);	$O(1)$

end while=0	<hr/>
	$O(1)$
enTodos(in c : campo in cr : conj(reg)) $\longrightarrow res$: bool	
$res \leftarrow \text{True};$	$O(1)$
$itcr \leftarrow \text{CrearItConj}(cr);$	$O(\#(cr))$
while haySiguiente(itcr) \wedge res do;	$O(\#(cr))$
if $\neg \text{pertenece}(c, \text{siguiente}(itcr))$ then;	$O(1)$
$res \leftarrow \text{false};$	$O(1)$
end if	
end while	<hr/>
	$O(\#(cr))$
combinarTodos(in c : campo in r_1 : reg in cr : conj(reg))	
$itcr \leftarrow \text{CrearItConjDic}(cr);$	$O(\#(cr))$
while haySiguiente(itcr) \wedge res do;	$O(\#(cr))$
if obtener($c, \text{siguiente}(itcr)$) == obtener(c, r) then;	$O(1)$
agregarCampos($r, \text{siguiente}(itcr)$);	$O(1)$
end if	
end while	<hr/>
	$O(\#(cr))$
unirRegistros(in c : campo in r_1 : reg in r_2 : reg) $\longrightarrow res$: reg	
$res \leftarrow nReg()O(1)\text{agregarCampos}(res, r_1);$	$O(1)$
agregarCampos(res, r_2);	$O(1)$
	<hr/>
	$O(\#(cr))$