

Algoritmos y Estructuras de Datos II

Departamento de Computación
Facultad de Ciencias Exactas y Naturales
Universidad de Buenos Aires

Trabajo Práctico II

Grupo: 12

Integrante	LU	Correo electrónico
Pondal, Iván	078/14	ivan.pondal@gmail.com
Paz, Maximiliano León	251/14	m4xileon@gmail.com
Mena, Manuel	313/14	manuelmena1993@gmail.com
Demartino, Francisco	348/14	demartino.francisco@gmail.com

Reservado para la cátedra

Instancia	Docente	Nota
Primera entrega		
Segunda entrega		

Índice

1. Módulo DCNet	3
1.1. Interfaz	3
1.1.1. Operaciones básicas de mapa	3
1.2. Representación	3
1.2.1. Representación de dcnet	3
2. Módulo Red	3
2.1. Interfaz	3

1. Módulo DCNet

1.1. Interfaz

se explica con: DCNET.

géneros: dcnet.

1.1.1. Operaciones básicas de mapa

CREAR() $\rightarrow res : dcnet$
Pre $\equiv \{true\}$
Post $\equiv \{res =_{obs} vacio()\}$
Complejidad: $O(1)$
Descripción: crea un mapa nuevo

1.2. Representación

1.2.1. Representación de dcnet

dcnet se representa con estr

donde estr es tupla(*topología*: red,
compusDCNet: vector(compuDCNet),
enEspera: dicc_{Trie}(puntero(compuDCNet)),
laQueMásEnvió: puntero(compuDCNet))

donde compuDCNet es tupla(*c*: compu,
buffer: conj(paquete),
encolados: colaPrioridad(nat, paqueteDCNet),
paqueteAEnviar: puntero(paqueteDCNet),
enviados: nat)

donde paqueteDCNet es tupla(*p*: paquete, *it*: itConj(paquete), *recorrido*: lista(compu))

2. Módulo Red

2.1. Interfaz

se explica con: RED.

géneros: red.

INICIARRED() $\rightarrow res : red$
Pre $\equiv \{true\}$
Post $\equiv \{res =_{obs} iniciaRed()\}$
Complejidad: $O(1)$
Descripción: crea una red nueva

AGREGARCOMPUTADORA(in/out *r*: red , in *c*: compu)
Pre $\equiv \{(\forall (: (c':compu)c' \in computadoras(r)) \Rightarrow ip(c) \neq ip(c')) \wedge (r = r_0) \}$
Post $\equiv \{r =_{obs} agregarComputadora(r_0, c) \}$
Complejidad: $O(L + n)$
Descripción: Agrega un computadora a la red

CONECTAR(in/out *r*: red , in *c*: compu , in *c'*: compu , in *i*: compu, in *i'*: compu)
Pre $\equiv \{ (r = r_0) \wedge ((c \in computadoras(r)) \wedge (c' \in computadoras(r))) \wedge (ip(c) \neq ip(c')) \}$

$\wedge (\neg \text{conectadas?}(r, c, c')) \wedge (\neg(\text{usaInterfaz?}(r, c, i)) \wedge \neg(\text{usaInterfaz?}(r, c', i'))))\}$

Post $\equiv \{r =_{\text{obs}} \text{conectar}(r_0, c, i, c', i')\}$

Complejidad: $O(L + n)$

Descripción: Conecta dos computadoras

COMPTUTADORAS(**in** r : red) $\rightarrow res$: conj(compu)

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

Post $\equiv \{res = \text{computadoras}(r)\}$

Complejidad: $O(1)$

CONECTADAS?(**in** r : red, **in** c : compu, **in** c' : compu) $\rightarrow res$: bool

Pre $\equiv \{(c \in \text{computadoras}(r)) \wedge (c' \in \text{computadoras}(r))\}$

Post $\equiv \{res = \text{conectadas?}(r, c, c')\}$

Complejidad: $O(1)$

INTERFAZUSADA(**in** r : red, **in** c : compu, **in** c' : compu) $\rightarrow res$: interfaz

Pre $\equiv \{\text{conectadas?}(r, c, c')\}$

Post $\equiv \{res = \text{interfazUsada}(r, c, c')\}$

Complejidad: $O(?)$

VECINOS(**in** r : red, **in** c : compu) $\rightarrow res$: conj(compu)

Pre $\equiv \{c \in \text{computadoras}(r)\}$

Post $\equiv \{res = \text{vecinos}(r, c)\}$

Complejidad: $O(n)$

USAINTERFAZ?(**in** r : red, **in** c : compu, **in** i : interfaz) $\rightarrow res$: bool

Pre $\equiv \{c \in \text{computadoras}(r)\}$

Post $\equiv \{res = \text{usaInterfaz}(r, c, i)\}$

Complejidad: $O(?)$

USAINTERFAZ?(**in** r : red, **in** c : compu, **in** i : interfaz) $\rightarrow res$: bool

Pre $\equiv \{c \in \text{computadoras}(r)\}$

Post $\equiv \{res = \text{usaInterfaz}(r, c, i)\}$

Complejidad: $O(n)$

CAMINOSMINIMOS(**in** r : red, **in** c : compu, **in** c' : compu) $\rightarrow res$: conj(secu(compu))

Pre $\equiv \{(c \in \text{computadoras}(r)) \wedge (c' \in \text{computadoras}(r))\}$

Post $\equiv \{res = \text{caminosMinimos}(r, c, i)\}$

Complejidad: $O(L)$

HAYCAMINO?(**in** r : red, **in** c : compu, **in** c' : compu) $\rightarrow res$: bool

Pre $\equiv \{(c \in \text{computadoras}(r)) \wedge (c' \in \text{computadoras}(r))\}$

Post $\equiv \{res = \text{hayCamino?}(r, c, i)\}$

Complejidad: $O(L)$