

Algoritmos y Estructuras de Datos II

Segundo Cuatrimestre de 2016

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

Trabajo Practico 2

Especificacion

Grupo De TP Algo2

Integrante	LU	Correo electrónico
Fernando Castro	627/12	fernandoarielcastro92@gmail.com
Philip Garrett	318/14	garrett.phg@gmail.com
Gabriel Salvo	564/14	gabrielsalvo.cap@gmail.com
Bernardo Tuso	792/14	btuso.95@gmail.com

Reservado para la cátedra

Instancia	Docente	Nota
Primera entrega		
Segunda entrega		

Índice

1. Modulos

Esta es un disenio(no tengo enie, paja) de la especificacion del Trabajo Practico 2 del 2^{do} cuatrimestre del 2016 presentada por la cathedra para la realizacion del Trabajo Practico 2. Ver enunciado:

<http://www.dc.uba.ar/materias/aed2/2016/2c/descargas/tps/tp2/view>

2. Módulo Coordenada

Interfaz

Representación

Coordenada se representa con `estr`

donde `estr` es `tupla(x: Nat, y: Nat)`

1) True

`Abs(e): estre - > Coor Rep(e)`

`c : Coor tq e.x = latitud(c) y e.y = longitud(c)`

3. Módulo Mapa

Interfaz

Representación

Mapa se representa con **estr**

donde **estr** es `tupla(coordenadas: ConjLineal, ancho: Nat)`

1) El ancho es igual al maximo de las coordenadas X.

1) $e.\text{Ancho} = \text{Max}(\text{TT1}(e.\text{Coordenadas}))$

$\text{Abs}(e)$: $\text{estre} \rightarrow \text{Mapa Rep}(e)$

$m : \text{Mapa} \text{ tq } e.\text{coordenadas} = \text{coordenadas}(m)$

4. Módulo Juego

Interfaz

usa: MAPA, COORDENADA.

se explica con: JUEGO.

géneros: juego.

CREARJUEGO(in m : mapa) $\rightarrow res$: juego

Pre $\equiv \{m = m_0\}$

Post $\equiv \{res =_{\text{obs}} \text{crearJuego}(m_0) \wedge \text{mapa}(res) =_{\text{obs}} m_0\}$

AGREGARPOKEMON(in/out j : juego, in c : coordenada, in p : pokemon) $\rightarrow res$: pokemon

Pre $\equiv \{j =_{\text{obs}} j_0\}$

Post $\equiv \{j =_{\text{obs}} \text{agregarPokemon}(res, c, j_0)\}$

Complejidad: $\Theta(|p| + EC * \log(EC))$

AGREGARJUGADOR(in/out j : juego) $\rightarrow res$: jugador

Pre $\equiv \{j =_{\text{obs}} j_0\}$

Post $\equiv \{j =_{\text{obs}} \text{agregarJugador}(res, j_0) \wedge \neg \text{estaConectado}(j, res) \wedge \neg \text{vacio?}(\text{pokemons}(j, res))\}$

Complejidad: $\Theta(J)$

Representación

Juego se representa con estr

donde **estr** es **tupla**(*pokemons*: diccTrie, *jugadores*: conjLineal , *jugadoresPorPosicion*: conjHash , *pokemonsPorPosicion*: conjHash , *jugadoresEnRango*: diccHeap , *mapa*: Mapa , *pT*: Nat)

Jugador se representa con jug

donde **jug** es **tupla**(*id*: Nat, *posicion*: Coordenada , *estaConectado*: Bool , *sanciones*: Nat , *pokeCapturados*: ConjLineal)

Pokemon se representa con poke

donde **poke** es **tupla**(*tipo*: String, *contador*: Nat , *posicion*: Coordenada , *salvaje*: Bool)

Rep: estr -> bool

- 1) La suma de toos los significados de pokemons es igual al PT
- 2) Todos las posiciones de jugPorPosicion esta contenida en el heap
- 3) Idem pokePorPosicion
- 4) Todo jugador que esta conectado y no expulsado, existe en jugPorPosicion
- 5) Para cada posicion hay un jugador en jugPorPosicion que pertenece a jugadores
- 6) Para cada pos en pokePorPosicion hay pokemon en pokemons
- 7) Para cada posicion en jugadoresEnRango, sus jugadores estan contenidos en jugadores
- 8) Para cada jugador en jugadores: si no esta expulsado, sus pokemons estan contenidos en pokemons del juego y no estan en pokemonsPorPosicion; y si esta conectado, su posicion pertenece al mapa del juego
- 9) Para cada pokemon en pokemons, si es salvaje: su contador es menor a 10, su posicion pertenece al mapa del juego y pertenece a pokemonEnPosicion

Abs(e): estre - > Jugo Rep(e)

pGo: Juego tq $e.\text{mapa} = \text{mapa}(pGo)$ y $e.\text{jugadores} = \text{jugadores}(pGo)$ y luego

(Para todo j : jugador) j pertenece $e.\text{jugadores}$ impluego

$j.\text{sanciones} = \text{sanciones}(j, pGo)$ ((j pertenece expulsados(pGo) y $j.\text{sanciones} \neq 10$)

oluego ($j.\text{pokesCapturados} = \text{pokemons}(j, pGo)$ y $j.\text{estaConectado} = \text{estaConectad}(j, pGo)$

y $j.\text{estaConectado}$ impluego $j.\text{pos} = \text{posicion}(j, pGo)$)) y

(Para todo p : pokemon) p pertenece $c.\text{pokemons}$ impluego (Para todo j : Jugador)

j pertenece $e.\text{jugadores}$ y luego p pertenece $\text{pokemons}(j, pGo)$ o [(Para todo c : coord)

c pertenece $e.\text{mapa.coordenadas}$ y luego $p = \text{pokemonEnPos}(c, pGo)$ y $\text{cantMovParaCap}(c, pGo)$

p.contador]

5. Módulo Tabla de Valores(*coordenada*, σ)

El módulo Tabla de Valores provee un diccionario por posiciones en el que se puede definir, borrar, y testear si hay un valor en una posición en tiempo $O(1)$.

El principal costo de pago al crear la estructura, dado de cuesta tiempo lineal *ancho* por *largo*.

Interfaz

parámetros formales

géneros *coordenada*, σ

se explica con: DICCIONARIO(κ, σ),

géneros: *tabla*(*coordenada*, σ).

Trabajo Practico 2 Operaciones básicas de tabla

VACÍO(**in** *Nat*: *a* *ncho*, **in** *Nat*: *l* *argo*) \rightarrow *res* : *tabla*(*coordenada*, σ)

Pre $\equiv \{\text{ancho} > 0 \wedge \text{largo} > 0\}$

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

Complejidad: $\Theta(\text{ancho} * \text{largo})$

Descripción: genera una tabla vacía.

DEFINIR(**in/out** *t*: *tabla*(*coordenada*, σ), **in** *c*: *coordenada*, **in** *s*: σ)

Pre $\equiv \{t =_{\text{obs}} t_0\}$

Post $\equiv \{t =_{\text{obs}} \text{definir}(t, c, s)\}$

Complejidad: $\Theta(1)$

Descripción: define el significado *s* en la tabla, en la posición representada por *c*.

Aliasing: Hay aliasing, pero no se como explicarlo TODO

DEFINIDO?(**in** *t*: *tabla*(*coordenada*, σ), **in** *c*: *coordenada*) \rightarrow *res* : *bool*

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

Post $\equiv \{res =_{\text{obs}} \text{def?}(t, c)\}$

Complejidad: $\Theta(1)$

Descripción: devuelve *true* si y sólo *c* tiene un valor en la tabla.

SIGNIFICADO(**in** *t*: *tabla*(*coordenada*, σ), **in** *c*: *coordenada*) \rightarrow *res* : σ

Pre $\equiv \{\text{def?}(t, c)\}$

Post $\equiv \{\text{alias}(res =_{\text{obs}} \text{Significado}(t, c))\}$

Complejidad: $\Theta(1)$

Descripción: devuelve el valor en la posición *c* de *t*.

BORRAR(**in/out** *t*: *tabla*(*coordenada*, σ), **in** *c*: *coordenada*)

Pre $\equiv \{t = t_0 \wedge \text{def?}(t, c)\}$

Post $\equiv \{t =_{\text{obs}} \text{borrar}(t_0, c)\}$

Complejidad: $\Theta(1)$

Descripción: elimina el valor en la posición *c* en *t*.