

Algoritmos

Algoritmos de Partida

iNuevaPartida(in $m : \text{mapa}$) $\rightarrow res : partida$

1: $partida \leftarrow \langle \rangle$	$\triangleright \mathcal{O}(1)$
2: $partida.mapa \leftarrow m$	$\triangleright \mathcal{O}(1)$
3: $partida.jugador \leftarrow mapa.inicio(m)$	$\triangleright \mathcal{O}(1)$
4: $i \leftarrow 0$	$\triangleright \mathcal{O}(1)$
5: while $i < mapa.cantChocolates(m)$ do	$\triangleright \mathcal{O}(c)$
6: $partida.chocolates[i] \leftarrow true$	$\triangleright \mathcal{O}(c)$
7: $i++$	$\triangleright \mathcal{O}(c)$
8: end while	
9: $partida.cantMov \leftarrow 0$	$\triangleright \mathcal{O}(1)$
10: $idChocolate \leftarrow mapa.idChocolate(m, partida.jugador)$	$\triangleright \mathcal{O}(1)$
11: $hayUnChocolate \leftarrow 0 \leq idChocolate < mapa.cantChocolates(m)$	$\triangleright \mathcal{O}(1)$
12: if $hayUnChocolate$ then	$\triangleright \mathcal{O}(1)$
13: $partida.chocolates[idChocolate] \leftarrow false$	$\triangleright \mathcal{O}(1)$
14: $partida.inmunidad \leftarrow 10$	$\triangleright \mathcal{O}(1)$
15: else	
16: $partida.inmunidad \leftarrow 0$	$\triangleright \mathcal{O}(1)$
17: end if	
18: $partida.perdio \leftarrow mapa.esCasilleroPeligroso(m, partida.jugador) \wedge partida.inmunidad = 0$	$\triangleright \mathcal{O}(1)$
19: $partida.gano \leftarrow false$	$\triangleright \mathcal{O}(1)$
20: $res \leftarrow partida$	$\triangleright \mathcal{O}(1)$

Complejidad: $\mathcal{O}(c)$

Justificación: La complejidad de este algoritmo es igual a recorrer el array $partida.chocolates$, que tiene longitud c , que es igual a la cantida de chocolates que contiene el mapa, para inicializar sus valores en $true$

iMover(in/out p : partida, in d : direccion)

1:	if $d = \text{ARRIBA}$ then	$\triangleright \mathcal{O}(1)$
2:	$\text{desplazamiento} \leftarrow (0, 1)$	$\triangleright \mathcal{O}(1)$
3:	end if	
4:	if $d = \text{ABAJO}$ then	$\triangleright \mathcal{O}(1)$
5:	$\text{desplazamiento} \leftarrow (0, -1)$	$\triangleright \mathcal{O}(1)$
6:	end if	
7:	if $d = \text{IZQUIERDA}$ then	$\triangleright \mathcal{O}(1)$
8:	$\text{desplazamiento} \leftarrow (-1, 0)$	$\triangleright \mathcal{O}(1)$
9:	end if	
10:	if $d = \text{DERECHA}$ then	$\triangleright \mathcal{O}(1)$
11:	$\text{desplazamiento} \leftarrow (1, 0)$	$\triangleright \mathcal{O}(1)$
12:	end if	
13:	$\text{nuevaPosicion} \leftarrow p.\text{jugador} + \text{desplazamiento}$	$\triangleright \mathcal{O}(1)$
14:	if $\neg \text{mapa.esPared}(p.\text{mapa}, \text{nuevaPosicion}) \wedge \text{mapa.enRango}(p.\text{mapa}, \text{nuevaPosicion})$ then	$\triangleright \mathcal{O}(1)$
15:	$p.\text{jugador} \leftarrow \text{nuevaPosicion}$	$\triangleright \mathcal{O}(1)$
16:	$p.\text{cantMov} + +$	$\triangleright \mathcal{O}(1)$
17:	$\text{idChocolate} \leftarrow \text{mapa.idChocolate}(p.\text{mapa}, p.\text{jugador})$	$\triangleright \mathcal{O}(1)$
18:	$\text{hayUnChocolate} \leftarrow 0 \leq \text{idChocolate} < \text{mapa.cantChocolates}(p.\text{mapa})$	$\triangleright \mathcal{O}(1)$
19:	if $\text{hayUnChocolate} \wedge_L p.\text{chocolates}[\text{idChocolate}]$ then	$\triangleright \mathcal{O}(1)$
20:	$p.\text{chocolates}[\text{idChocolate}] \leftarrow \text{false}$	$\triangleright \mathcal{O}(1)$
21:	$p.\text{inmunidad} \leftarrow 10$	$\triangleright \mathcal{O}(1)$
22:	else	
23:	$p.\text{inmunidad} \leftarrow \max(p.\text{inmunidad} - 1, 0)$	$\triangleright \mathcal{O}(1)$
24:	end if	
25:	$p.\text{partida.perdido} \leftarrow \text{mapa.esCasilleroPeligroso}(m, p.\text{partida.jugador}) \wedge p.\text{partida.inmunidad} = 0$	$\triangleright \mathcal{O}(1)$
26:	$p.\text{partida.gano} \leftarrow \text{mapa.llegada}(m) = p.\text{partida.jugador}$	$\triangleright \mathcal{O}(1)$
27:	end if	

Complejidad: $\mathcal{O}(1)$

Justificación: No se recorre ninguna array, solo se accede a índices específicos. Los datos que provee el mapa, se encuentran guardados en arrays, que se complejidad de acceso es $\mathcal{O}(1)$, igualmente el acceso y modificación del array $p.\text{chocolates}$ es $\mathcal{O}(1)$

iGanó?(in p : partida) $\rightarrow res : \text{bool}$

1:	$res \leftarrow p.\text{gano}$	$\triangleright \Theta(1)$
----	--------------------------------	----------------------------

Complejidad: $\Theta(1)$

iPerdió?(in p : partida) $\rightarrow res : \text{bool}$

1:	$res \leftarrow p.\text{perdido}$	$\triangleright \Theta(1)$
----	-----------------------------------	----------------------------

Complejidad: $\Theta(1)$

iJugador(in p : partida) $\rightarrow res : \text{coordenada}$

1:	$res \leftarrow p.\text{jugador}$	$\triangleright \Theta(1)$
----	-----------------------------------	----------------------------

Complejidad: $\Theta(1)$

iCantMov(in p : partida) $\rightarrow res : nat$ $_1: res \leftarrow p.cantMov$ $\triangleright \Theta(1)$ Complejidad: $\Theta(1)$
