

Algoritmos

Algoritmos de Mapa

```

iNuevoMapa(in largo: nat, in alto: nat, in inicio: coordenada, in llegada: coordenada, in fantasmas:
conj (coordenada), in paredes: conj (coordenada), in chocolates: conj (coordenada))  $\rightarrow$  res : mapa
  1: int contadorDeChocolate  $\leftarrow$  0  $\triangleright \mathcal{O}(1)$ 
  2: mapa  $\leftarrow$   $\langle \rangle$   $\triangleright \mathcal{O}(1)$ 
  3: mapa.inicio  $\leftarrow$  inicio  $\triangleright \mathcal{O}(1)$ 
  4: mapa.llegada  $\leftarrow$  llegada  $\triangleright \mathcal{O}(1)$ 
  5: mapa.largo  $\leftarrow$  largo  $\triangleright \mathcal{O}(1)$ 
  6: mapa.alto  $\leftarrow$  alto  $\triangleright \mathcal{O}(1)$ 
  7: mapa.#chocolates  $\leftarrow$  chocolates.longitud  $\triangleright \mathcal{O}(1)$ 
  8: for int x = 0; x < mapa.largo; x ++ do  $\triangleright \mathcal{O}(largo)$ 
  9:   for int y = 0; y < mapa.alto; y ++ do  $\triangleright \mathcal{O}(largo * alto)$ 
 10:     mapa.matriz[x][y].pared = (x, y) in paredes  $\triangleright \mathcal{O}(largo * alto * |P|)$ 
 11:     mapa.matriz[x][y].fantasma = (x, y) in fantasmas  $\triangleright \mathcal{O}(largo * alto * |F|)$ 
 12:     mapa.matriz[x][y].peligorsa = seAsusta((x, y), fantasmas)  $\triangleright \mathcal{O}(largo * alto * |F|)$ 
 13:     if (x, y) in chocolates then  $\triangleright \mathcal{O}(largo * alto * |C|)$ 
 14:       mapa.matriz[x][y].idChocolate  $\leftarrow$  contadorDeChocolate  $\triangleright \mathcal{O}(largo * alto * |C|)$ 
 15:       contadorDeChocolate ++  $\triangleright \mathcal{O}(largo * alto * |C|)$ 
 16:     else
 17:       mapa.matriz[x][y].idChocolate  $\leftarrow$  -1  $\triangleright \mathcal{O}(largo * alto * |C|)$ 
 18:     end if
 19:   end for
 20: end for
 21: res  $\leftarrow$  mapa  $\triangleright \mathcal{O}(1)$ 

```

Complejidad: $\mathcal{O}((alto \cdot largo \cdot (\#chocolates + \#fantasmas + \#paredes + 1)))$

seAsusta(in c : coordenada, in $fantasmas$: conj(coordenada) out res : bool)

```

1:  $res \leftarrow False$ 
2:  $int\ i \leftarrow 0$   $\triangleright \mathcal{O}(1)$ 
3: while  $i < fantasma.longitud$  do  $\triangleright \mathcal{O}(|f|)$ 
4:   if  $distancia(c, fantasma[i]) \leq 3$  then  $\triangleright \mathcal{O}(|f|)$ 
5:      $res \leftarrow true$   $\triangleright$  si es el peor caso no entra
6:   else
7:      $i++$   $\triangleright \mathcal{O}(|f|)$ 
8:   end if
9: end while

```

Complejidad: $\mathcal{O}(|f|)$

distancia(in c : coordenada, in f : coordenada out res : nat)

```

1:  $res \leftarrow valorAbsoluto(c_1 - f_1) + valorAbsoluto(c_2 - f_2)$   $\triangleright \Theta(1)$ 

```

Complejidad: $\Theta(1)$

iesCasilleroPeligroso(in m : mapa, in $posicion$: coordenada) $\rightarrow res$: bool

```

1: if  $enRango(m, posicion) \wedge_L m.matriz[posicion_1][posicion_2].peligrosa$  then  $\triangleright \Theta(1)$ 
2:    $res \leftarrow true$   $\triangleright \Theta(1)$ 
3: else
4:    $res \leftarrow false$   $\triangleright \Theta(1)$ 
5: end if

```

Complejidad: $\Theta(1)$

ienRango(in m : mapa, in $posicion$: coordenada) $\rightarrow res$: bool

```

1:  $res \leftarrow (0 \leq posicion_1 < m.alto \wedge_L 0 \leq posicion_2 < m.largo)$   $\triangleright \Theta(1)$ 

```

Complejidad: $\Theta(1)$

iCantChocolates(in map : mapa) $\rightarrow res$: nat

```

1:  $res \leftarrow map.\#chocolates$   $\triangleright \Theta(1)$ 

```

Complejidad: $\Theta(1)$

iEsPared(in map : mapa, in $posicion$: coordenada) $\rightarrow res$: bool

```

1:  $res \leftarrow map.matriz[posicion_1][posicion_2].pared$   $\triangleright \Theta(1)$ 

```

Complejidad: $\Theta(1)$

iInicio(in map : mapa) $\rightarrow res$: coordenada

```

1:  $res \leftarrow map.inicio$   $\triangleright \Theta(1)$ 

```

Complejidad: $\Theta(1)$

iLlegada(in $map : \text{mapa}$) $\rightarrow res : \text{coordenada}$

$_1: res \leftarrow map.llegada$

$\triangleright \Theta(1)$

Complejidad: $\Theta(1)$

iIdChocolate(in $m : \text{mapa}$, in $posicion : \text{coordenada}$) $\rightarrow res : \text{nat}$

$_1: res \leftarrow m.matriz.[posicion_1].[posicion_2].idChocolate$

$\triangleright \Theta(1)$

Complejidad: $\Theta(1)$
