DEPARTAMENTO
DE COMPUTACION
Facultad de Ciencias Exactas y Naturales - UBA

Trabajo Práctico Número 1

Especificación

Algoritmos y Estructuras de Datos I

Grupo: 4

Integrante	LU	Correo electrónico
Aun Castells, María Virginia	366/13	`vauncastells@hotmail.com`
Motta, Leandro	85/14	`leamotta@msn.com`
Zdanovitch, Nikita	520/14	`3hb.tch@gmail.com`
de Monasterio, Francisco	764/13	`franciscodemonasterio@outlook.com`

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) 4576-3359
http://www.fcen.uba.ar

1. Resolución

Ejercicio 1. Blur:

```
problema blur($imagen : [[(\mathbb{Z},\mathbb{Z},\mathbb{Z})]],\ k : \mathbb{Z}) = res : [[(\mathbb{Z},\mathbb{Z},\mathbb{Z})]]$ {
    requiere : $k > 0$;
    requiere : $esImagenValida(imagen)$;
    asegura : $mismoTamano(imagen,\ res)$;
    asegura : $(\forall\ y \leftarrow [0\mathinner{..}alto(imagen)))\ (\forall\ x \leftarrow [0\mathinner{..}ancho(imagen)))$
        if $esKCompleto(kVecinos(imagen,\ y,\ x,\ k),\ k)$
        then $esPromedio(res, imagen,\ y,\ x,\ k)$
        else $esNegro(res,\ y,\ x)$;
}
```

Ejercicio 2. Acuarela:

```
problema acuarela($imagen : [[(\mathbb{Z},\mathbb{Z},\mathbb{Z})]],\ k : \mathbb{Z}) = res : [[(\mathbb{Z},\mathbb{Z},\mathbb{Z})]]$ {
    requiere : $k > 0$;
    requiere : $esImagenValida(imagen)$;
    asegura : $mismoTamano(imagen,\ res)$;
    asegura : $(\forall\ i \leftarrow [0\mathinner{..}alto(res)),\ j \leftarrow [0\mathinner{..}ancho(res)))$
        $res[i][j] == medianaONegro(i,\ j,\ img,\ k)$;
}
```

Ejercicio 3. Dividir:

```
problema dividir($imagen : [[(\mathbb{Z},\mathbb{Z},\mathbb{Z})]],\ m, n : \mathbb{Z}) = res : [[[(\mathbb{Z},\mathbb{Z},\mathbb{Z})]]]$ {
    requiere : $0 < n < alto(imagen)$;
    requiere : $0 < m < ancho(imagen)$;
    requiere : $esImagenValida(imagen)$;
    requiere : $tieneSuperficie(imagen)$;
    requiere divideEnFilasIguales: $alto(imagen) \bmod m == 0$;
    requiere divideEnColumnasIguales: $ancho(imagen) \bmod n == 0$;
    asegura : $mismos\,(res,\ separarHorizontal\,(separarVertical(imagen,\ n),\ m))$;
}
```

Ejercicio 4. Pegar:

```
problema pegar($imagen : [[(\mathbb{Z},\mathbb{Z},\mathbb{Z})]],\ color : (\mathbb{Z},\mathbb{Z},\mathbb{Z}),\ parche : [[(\mathbb{Z},\mathbb{Z},\mathbb{Z})]]) =$ {
    modifica $imagen$;
    requiere : $esImagenValida(\mathsf{pre}(imagen))$;
    requiere : $esImagenValida(parche)$;
    requiere : $tieneSuperficie(parche)$;
    requiere : $alto(parche) \leq alto(imagen)\ \wedge\ ancho(parche) \leq ancho(imagen)$;
    requiere : $esByte(color)$;
    asegura : $mismoTamano(imagen,\ \mathsf{pre}(imagen))$;
    asegura : if $existeDestino(\mathsf{pre}(imagen),\ parche, color)$
        then $estaPegado(parche,\ \mathsf{pre}(imagen),\ imagen,\ destino(\mathsf{pre}(imagen),\ parche,\ color))$
        else $imagen == \mathsf{pre}(imagen)$
}
```

1.1. Auxiliares

- aux $esImagenValida(img : [[(\mathbb{Z},\mathbb{Z},\mathbb{Z})]])$: Bool = $esRectangular(imagen)\ \wedge$ $(\forall\ f \leftarrow imagen)(\forall\ px \leftarrow f)\ esPixelValido(px)$;

- aux $mismoTamano(img, img2 : [[(\mathbb{Z},\mathbb{Z},\mathbb{Z})]]) : \mathsf{Bool} = alto(img) == alto(img2) \land ancho(img) == ancho(img2)$;
- aux $tieneSuperficie(img : [[(\mathbb{Z},\mathbb{Z},\mathbb{Z})]]) : \mathsf{Bool} = alto(img) > 0 \ \land \ ancho(img) > 0$;
- aux $esRectangular(img : [[(\mathbb{Z},\mathbb{Z},\mathbb{Z})]]) : \mathsf{Bool} = (\forall a \leftarrow img)\ |a| == |\mathsf{cab}(img)|$;
- aux $esPixelValido(px : (\mathbb{Z},\mathbb{Z},\mathbb{Z})) : \mathsf{Bool} = esByte(\mathsf{prm}(px)) \land esByte(\mathsf{sgd}(px)) \land esByte(\mathsf{trc}(px))$;
- aux $esByte(b : \mathbb{Z}) : \mathsf{Bool} = 0 \leq b \leq 255$;
- aux $alto(img : [[(\mathbb{Z},\mathbb{Z},\mathbb{Z})]]) : \mathbb{Z} = |img|$;
- aux $ancho(img : [[(\mathbb{Z},\mathbb{Z},\mathbb{Z})]]) : \mathbb{Z} = \ \mathsf{if}\ |img| == 0\ \mathsf{then}\ 0\ \mathsf{else}\ |\mathsf{cab}(img)|$;
- aux $esKCompleto(kv : [(\mathbb{Z},\mathbb{Z},\mathbb{Z})],\ k : \mathbb{Z}) : \mathsf{Bool} = |kv| == (k + k - 1)^2$;
- aux $esPromedio(res, img : [[(\mathbb{Z},\mathbb{Z},\mathbb{Z})]],\ y, x, k : \mathbb{Z}) : \mathsf{Bool} =$
  $\mathsf{prm}(pixel(res,\ y,\ x)) == (\mathsf{sum}([\mathsf{prm}(p) \mid p \leftarrow kVecinos(img,\ y,\ x,\ k)])\ /\ |kVecinos(img,\ y,\ x,\ k)|) \land$
  $\mathsf{sgd}(pixel(res,\ y,\ x)) == (\mathsf{sum}([\mathsf{sgd}(p) \mid p \leftarrow kVecinos(img,\ y,\ x,\ k)])\ /\ |kVecinos(img,\ y,\ x,\ k)|) \land$
  $\mathsf{sgd}(pixel(res,\ y,\ x)) == (\mathsf{sum}([\mathsf{trc}(p) \mid p \leftarrow kVecinos(img,\ y,\ x,\ k)])\ /\ |kVecinos(img,\ y,\ x,\ k)|)$;
- aux $esNegro(img : [[(\mathbb{Z},\mathbb{Z},\mathbb{Z})]],\ y, x : \mathbb{Z}) : \mathsf{Bool} = \mathsf{prm}(pixel(img,\ y,\ x)) == 0 \land \mathsf{sgd}(pixel(img,\ y,\ x)) == 0 \ \land\ \mathsf{trc}(pixel(img,\ y,\ x)) == 0$;
- aux $pixel(img : [[(\mathbb{Z},\mathbb{Z},\mathbb{Z})]],\ y, x : \mathbb{Z}) : (\mathbb{Z},\mathbb{Z},\mathbb{Z}) = \ \mathsf{if}\ esIndiceValido(y,\ x,\ img)\ \mathsf{then}\ img[y][x]\ \mathsf{else}\ (0,\ 0,\ 0)$;
- aux $kVecinos(img : [[(\mathbb{Z},\mathbb{Z},\mathbb{Z})]],\ y, x, k : \mathbb{Z}) : [(\mathbb{Z},\mathbb{Z},\mathbb{Z})] = [pixel(img,\ \mathsf{prm}(c),\ \mathsf{sgd}(c)) \mid c \leftarrow kIndices(y,\ x,\ k),\ esIndiceValido(\mathsf{prm}(c),\ \mathsf{sgd}(c),\ img)]$;
- aux $kIndices(y, x, k : \mathbb{Z}) : [(\mathbb{Z},\mathbb{Z})] = [(i,\ j) \mid i \leftarrow (y - k \mathinner{..} y + k),\ j \leftarrow (x - k \mathinner{..} x + k)]$;
- aux $esIndiceValido(y, x : \mathbb{Z},\ img : [[(\mathbb{Z},\mathbb{Z},\mathbb{Z})]]) : \mathsf{Bool} = 0 \leq y < alto(img) \land 0 \leq x < ancho(img)$;
- aux $medianaONegro(i, j : \mathbb{Z},\ img : [[(\mathbb{Z},\mathbb{Z},\mathbb{Z})]],\ k : \mathbb{Z}) : (\mathbb{Z},\mathbb{Z},\mathbb{Z}) =$
  $\mathsf{if}\ esKCompleto(kVecinos(img,\ i,\ j,\ k),\ k)$
  $\mathsf{then}\ mediana(kVecinos(img,\ i,\ j,\ k))$
  $\mathsf{else}\ (0,\ 0,\ 0)$;
- aux $mediana(kv : [(\mathbb{Z},\mathbb{Z},\mathbb{Z})]) : (\mathbb{Z},\mathbb{Z},\mathbb{Z}) =$
  $(valorMediana([\mathsf{prm}(a) \mid a \leftarrow kv]),$
  $valorMediana([\mathsf{sgd}(a) \mid a \leftarrow kv]),$
  $valorMediana([\mathsf{trc}(a) \mid a \leftarrow kv]))$;
- aux $valorMediana(xs : [\mathbb{Z}]) : \mathbb{Z} = enOrden(xs)[|xs|/2]$;
- aux $enOrden(xs : [\mathbb{Z}]) : [\mathbb{Z}] = [x \mid i \leftarrow [0 \mathinner{..} |xs|),\ x \leftarrow xs,\ cuentaMenores(xs,\ x) == i]$;
- aux $cuentaMenores(xs : [\mathbb{Z}],\ x : \mathbb{Z}) : \mathbb{Z} = |[1 \mid y \leftarrow xs,\ y < x]|$;
- aux $separarVertical(img : [[(\mathbb{Z},\mathbb{Z},\mathbb{Z})]],\ columnas : \mathbb{Z}) : [[[(\mathbb{Z},\mathbb{Z},\mathbb{Z})]]] =$
  $\big[verticalizarImagen(img,\ columnas)[alto(img)i \mathinner{..} alto(img)(i+1)) \ \big|\ i \leftarrow [0 \mathinner{..} columnas)\big]$;
- aux $verticalizarImagen(img : [[(\mathbb{Z},\mathbb{Z},\mathbb{Z})]],\ columnas : \mathbb{Z}) : [[(\mathbb{Z},\mathbb{Z},\mathbb{Z})]] =$
  $\big[img[i][Ancho(img)k/columnas \mathinner{..} Ancho(img)(k+1)/columnas) \ \big|\ k \leftarrow [0 \mathinner{..} columnas),\ i \leftarrow [0 \mathinner{..} Alto(img))\big]$;
- aux $separarHorizontal(listaimg : [[[(\mathbb{Z},\mathbb{Z},\mathbb{Z})]]],\ filas : \mathbb{Z}) : [[[(\mathbb{Z},\mathbb{Z},\mathbb{Z})]]] =$
  $\big[listaimg[i][|cab(listaimg)|k/filas \mathinner{..} |cab(listaimg)|(k{+}1)/filas) \ \big|\ k \leftarrow [0 \mathinner{..} filas),\ i \leftarrow [0 \mathinner{..} |listaimg|)\big]$;
- aux $cuenta(x : T,\ a : [T]) : \mathbb{Z} = \big|[y \mid y \leftarrow a,\ y == x]\big|$;
- aux $mismos(a, b : [T]) : \mathsf{Bool} = (|a| == |b|) \land (\forall\ c \leftarrow a)\ cuenta(c,\ a) == cuenta(c,\ b)$;

- aux $existeDestino(img, parche : [[(\mathbb{Z},\mathbb{Z},\mathbb{Z})]],\ color : (\mathbb{Z},\mathbb{Z},\mathbb{Z})) : \mathsf{Bool} =$
  $|posiblesDestinos(img,\ parche,\ color)| == 1\ \wedge$
  $noExistenPuntosAfuera\big(img,\ color,\ \mathsf{cab}(posiblesDestinos(img,\ parche,\ color))\big);$

- aux $noExistenPuntosAfuera\big(img : [[(\mathbb{Z},\mathbb{Z},\mathbb{Z})]],\ color : (\mathbb{Z},\mathbb{Z},\mathbb{Z}),\ r : ((\mathbb{Z},\mathbb{Z}),\ (\mathbb{Z},\mathbb{Z}))\big) : \mathsf{Bool} =$
  $(\forall\ yx \leftarrow obtengoPuntos(img,\ color))$
  $(y(r) \leq \mathsf{pre}(yx) < y(r) + h(r)\ \wedge\ x(r) \leq \mathsf{sgd}(yx) < x(r) + w(r));$

- aux $posiblesDestinos(img, parche : [[(\mathbb{Z},\mathbb{Z},\mathbb{Z})]],\ color : (\mathbb{Z},\mathbb{Z},\mathbb{Z})) : [((\mathbb{Z},\mathbb{Z}),\ (\mathbb{Z},\mathbb{Z}))] =$
  $\big[r \mid r \leftarrow posiblesRectangulos(img,\ parche),\ esDeColor(img,\ r,\ color)\big]$

- aux $posiblesRectangulos(img, parche : [[(\mathbb{Z},\mathbb{Z},\mathbb{Z})]]) : [((\mathbb{Z},\mathbb{Z}),\ (\mathbb{Z},\mathbb{Z}))] =$
  $\big[rect(y,\ x,\ alto(parche),\ ancho(parche))$
  $\big|\ y \leftarrow [0\mathinner{..}alto(img) - alto(parche)),\ x \leftarrow [0\mathinner{..}ancho(img) - ancho(parche))];$

- aux $esDeColor\big(img : [[(\mathbb{Z},\mathbb{Z},\mathbb{Z})]],\ r : ((\mathbb{Z},\mathbb{Z}),\ (\mathbb{Z},\mathbb{Z})),\ color : (\mathbb{Z},\mathbb{Z},\mathbb{Z})\big) : \mathsf{Bool} =$
  $\big(\forall\ y \leftarrow [y(r)\mathinner{..}y(r) + h(r)),\ x \leftarrow [x(r)\mathinner{..}x(r) + w(r))\ \big)\ pixel(img,\ y,\ x) == color;$

- aux $obtengoPuntos\big(img : [[(\mathbb{Z},\mathbb{Z},\mathbb{Z})]],\ color : (\mathbb{Z},\mathbb{Z},\mathbb{Z})\big) : [(\mathbb{Z},\mathbb{Z})] =$
  $\big[\ (y,\ x) \mid y \leftarrow [0\mathinner{..}alto(img)),\ x \leftarrow [0\mathinner{..}ancho(img)),\ img[y][x] == color\big];$

- aux $destino(img, parche : [[(\mathbb{Z},\mathbb{Z},\mathbb{Z})]],\ color : (\mathbb{Z},\mathbb{Z},\mathbb{Z})) : ((\mathbb{Z},\mathbb{Z}),\ (\mathbb{Z},\mathbb{Z})) =$
  if $existeDestino(img,\ parche,\ color)$
  then $\mathsf{cab}(posiblesDestinos(img,\ parche,\ color))$
  else $((0,\ 0),\ (0,\ 0));$

- aux $estaPegado\big(parche, preimg, img : [[(\mathbb{Z},\mathbb{Z},\mathbb{Z})]],\ r : ((\mathbb{Z},\mathbb{Z}),\ (\mathbb{Z},\mathbb{Z}))\big) : \mathsf{Bool} =$
  $\big(\forall\ i \leftarrow [0\mathinner{..}Alto(img)),\ j \leftarrow [0\mathinner{..}Ancho(img)\big)$
  if $\big(prm(pos) \leq i < prm(pos) + alto(parche)\ \wedge\ seg(pos) \leq j < seg(pos) + ancho(parche)\big)$
  then $img[i][j] == parche[i - prm(pos)][j - seg(pos)];$
  else $img[i][j] == preimg[i][j];$

- aux $rect(y, x, h, w : \mathbb{Z}) : ((\mathbb{Z},\mathbb{Z}),\ (\mathbb{Z},\mathbb{Z})) = ((y,\ x),\ (h,\ w));$

- aux $y(rect : ((\mathbb{Z},\mathbb{Z}),\ (\mathbb{Z},\mathbb{Z}))) : \mathbb{Z} = \mathsf{prm}(\mathsf{prm}(rect));$

- aux $x(rect : ((\mathbb{Z},\mathbb{Z}),\ (\mathbb{Z},\mathbb{Z}))) : \mathbb{Z} = \mathsf{prm}(\mathsf{sgd}(rect));$

- aux $h(rect : ((\mathbb{Z},\mathbb{Z}),\ (\mathbb{Z},\mathbb{Z}))) : \mathbb{Z} = \mathsf{sgd}(\mathsf{prm}(rect));$

- aux $w(rect : ((\mathbb{Z},\mathbb{Z}),\ (\mathbb{Z},\mathbb{Z}))) : \mathbb{Z} = \mathsf{sgd}(\mathsf{sgd}(rect));$