Problema C

Ricardo Enrique González

Camilo Zambrano

201425733

201515438

1. El algoritmo cuenta con tres ciclos. El primero, es encargado de repetirse para tomar la siguiente imagen de entrada con sus tamaños y los otros dos se encargan de tomar los tamaños y la imagen para calcular la respuesta de esa imagen en cuestión. En general el primer ciclo no es tan importante, solo se ejecuta una vez para cada imagen de la entrada, en cambio, los otros dos ciclos son los que más peso tienen en la ejecución del programa y los que al final permiten calcular si la mayor cantidad de pixeles blancos seguidos se encontraban en una fila (“H” en la salida) o en una columna (“V” en la salida). A continuación, se provee el código en GCL de los dos ciclos utilizados en el cálculo de la respuesta en donde los píxeles de entrada son vistos como una matriz de m filas y n columnas.

[| {CTX: B[0..m,0..n]:{1⋀0} , m,n:nat}

actual,maxC,maxV,i,j:= 0,0,0,0,0;

V[0..n,0..1]:=0; {vector para guardar la cantidad de 1’s en cada columna.}

{Inv I1:0<i<m ⋀ 0<j<=n, b[i,j] se está procesando y es necesario procesar lo que queda de b[0..m,0..n]}

{Cota: m-i}

do i<m ⋀ j<=n ⟶ if j<n⟶ if b[i,j]=1 ⟶actual,v[j,1],j:=actual+1,v[j,1]+1,j+1;

[] b[i,j]=0⟶if v[j,0]<v[j,1]⟶v[j,0]:=v[j,1] fi v[j,0]:=0;

if maxC<actual⟶maxC:=actual fi actual,j:=0,0;

if

[] j=n⟶if maxC<actual⟶maxC:=actual fi actual,j,i:=0,0,i+1;

fi

od

{R1: maxC ⋀ v[0...n,0..1]:nat}

i:=0;

{Inv I2: 0<i<n, maxV:=(maxV,max(v[i,0],v[i,1]))}

{Cota: n-i}

do i<m ⟶ if maxV<v[i,0] ⋀ v[i,1]<v[i,0] ⟶maxV:=v[j,0];

[] maxV<v[i,1] ⋀ v[i,0]<v[i,1] ⟶maxV:=v[j,1];

if

od

{R2: maxV ≠0}

if maxC>=maxV ⟶r:=’H’+maxC;

[] maxV>=maxC ⟶r:=’V’+maxV;

fi

{R: respuesta = ‘H’+maxC ∨ ‘V’+maxV}

|]

Como se puede ver el algoritmo hace un ciclo en el que se va