

En la clase vértice

ptoArt (linkedList<vertice> puntos, int[] cont)

Comienzo

```
    cont[0] ++
    this.numBp  $\leftarrow$  cont[0]
    this.numBajo  $\leftarrow$  cont[0]
    linkedList<vertice> hijos
    Para cada adyacencia en this.adyacencias
        adyacente  $\leftarrow$  adyacencia.destino
        Si no(adyacente.visitado) entonces
            adyacente.ptosArt(puntos, cont)
            hijos.add(adyacente)
            this.numBajo  $\leftarrow$  mínimoEntre (this.numBajo, adyacente.numBajo)
        sino
            this.numBajo  $\leftarrow$  mínimoEntre (this.numBajo, adyacente.numBp)
    finsi
    fin para cada
    Si this. numBp > 1 entonces
        Para cada hijo en hijos hacer
            Si hijo.numBajo  $\geq$  this.numBp entonces
                puntos.add(this)
            fin si
        fin para cada
    Sino
        Si hijos.largo > 1 entonces
            puntos.add(this)
        fin si
    fin si
```

Fin

En la clase vértice

ptoArt (linkedList<vertice> puntos, ref int prof)

Comienzo

```
    prof++
    this.numBp  $\leftarrow$  prof
    this.numBajo  $\leftarrow$  prof
    linkedList<vertice> hijos
    Para cada adyacencia en this.adyacencias
        adyacente  $\leftarrow$  adyacencia.destino
        Si no(adyacente.visitado) entonces
            adyacente.ptosArt(puntos, prof)
            hijos.add(adyacente)
            this.numBajo  $\leftarrow$  mínimoEntre (this.numBajo, adyacente.numBajo)
        sino
            this.numBajo  $\leftarrow$  mínimoEntre (this.numBajo, adyacente.numBp)
    finsi
    fin para cada
    Si this. numBp > 1 entonces
        Para cada hijo en hijos hacer
            Si hijo.numBajo  $\geq$  this.numBp entonces
                puntos.add(this)
            fin si
        fin para cada
    Sino
        Si hijos.largo > 1 entonces
            puntos.add(this)
        fin si
    fin si
```

Fin