

---

**Algoritmo 1: RGCLI**

---

**Input:** número de  $k$  vecinos más cercanos  $k_e$ , número de RGCLI  
vecinos más cercanos  $k_i$ , lista de datos etiquetados  $L$ , *dataset*  
 $X$

**Output:** Grafo  $G$

```
1  $V \leftarrow$  vértices a partir de  $X$ 
2  $E, W \leftarrow \emptyset$ 
3  $G \leftarrow (V, E, W)$ 
4  $kdtree \leftarrow$  a partir de  $X$ 
5  $kNN \leftarrow$  dic
6  $F \leftarrow$  dic
7  $L \leftarrow$  dic
8 BúsquedaKNN()
9 BúsquedaRGCLI()
10 return  $G$ 

11 Function BúsquedaKNN():
12   for  $v_i \in V$ 
13      $kNN[v_i] \leftarrow kdtree.query(v_i, k_e)$ 
14      $L[v_i] \leftarrow$  encontrar punto etiquetado más cercano en  $L$ 
15      $F[v_i] \leftarrow$  encontrar el  $k$ -ésimo vecino más lejano de  $v_i$ 
16   endfor

17 Function BúsquedaRGCLI():
18   for  $v_i \in V$ 
19      $\epsilon \leftarrow$  dic
20     for  $v_j \in kNN[v_i]$ 
21       if  $dist(v_i, v_j) \leq dist(v_j, F[v_j])$ 
22          $e \leftarrow (v_j, v_i)$ 
23          $\epsilon[e] \leftarrow dist(v_i, v_j) + dist(v_j, L[v_j])$ 
24       end
25     endfor
26   endfor
27    $E^* \leftarrow$  obtener  $k_i$  aristas con la menor puntuación de  $\epsilon$ 
28    $E \leftarrow E \cup E^*$ 
29    $w(e) \leftarrow 1 \forall e = (v_i, v_j) \in E^*$ 
```

---