

```

1  -----
2  -- Estructuras de Datos. 2018/19
3  -- 2º Curso del Grado en Ingeniería [Informática | del Software | de Computadores].
4  -- Escuela Técnica Superior de Ingeniería en Informática. UMA
5  --
6  -- Examen 4 de febrero de 2019
7  --
8  -- ALUMNO/NAME:
9  -- GRADO/STUDIES:
10 -- NÚM. MÁQUINA/MACHINE NUMBER:
11 --
12 -----
13
14 module Kruskal(kruskal, kruskals) where
15
16 import qualified DataStructures.Dictionary.AVLDictionary as D
17 import qualified DataStructures.PriorityQueue.LinearPriorityQueue as Q
18 import DataStructures.Graph.DictionaryWeightedGraph
19
20 kruskal :: (Ord a, Ord w) => WeightedGraph a w -> [WeightedEdge a w]
21 kruskal wg = alg (dic (vertices wg)) (pq (edges wg)) []
22   where
23     dic [] = D.empty
24     dic (x:xs) = D.insert x x (dic xs)
25
26     pq [] = Q.empty
27     pq (x:xs) = Q.enqueue x (pq xs)
28
29     alg d p t
30       | Q.isEmpty p = t
31       | otherwise = alg (checkDic d (Q.first p)) (Q.dequeue p) (checkT t d
32 (Q.first p))
33
34     checkDic d (WE src w dst)
35       | representante d src /= representante d dst = D.insert (representante d
36 dst) src d
37       | otherwise = d
38
39     checkT t d ed@(WE src w dst)
40       | representante d src /= representante d dst = ed : t
41       | otherwise = t
42
43     representante dic elem = u
44       where
45         Just u = D.valueOf elem dic
46
47 -- Solo para evaluación continua / only for part time students
48 kruskals :: (Ord a, Ord w) => WeightedGraph a w -> [[WeightedEdge a w]]
49 kruskals = undefined

```