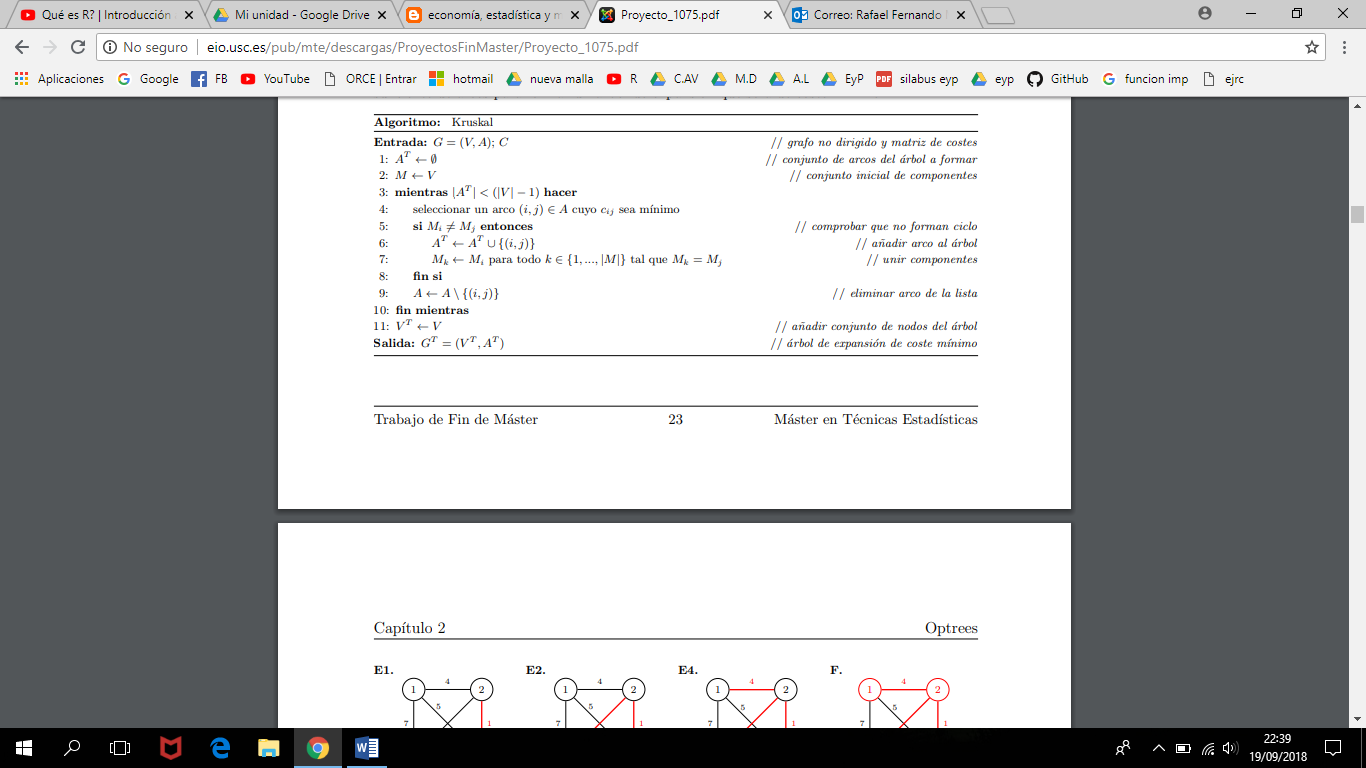
Algoritmo de Kruskal

(descripción de objetos y funciones)

No hay que ir demasiado lejos para encontrarnos con un segundo algoritmo capaz de resolver el problema de los árboles de coste mínimo. Su creación se debe a Joseph Kruskal, compañero de Prim en Bell Labs, quien lo propuso en 1956.

Conocido desde entonces por su apellido, la idea principal del algoritmo de Kruskal se resume en conectar arcos, y no nodos, secuencialmente. De esta forma hace innecesaria la elección de un nodo de partida. Empezando con el conjunto inicial de arcos, el algoritmo de Kruskal selecciona aquel con el menor coste, uno cualquiera si hay varios, y lo incorpora al árbol. A partir de ahí repite dicho paso en cada etapa, asegurándose de que no se producen ciclos, hasta que logra añadir el número suficiente de arcos para formar un árbol de expansión que será de coste mínimo.

(técnica a utilizar)



El algoritmo de Kruskal puede ser portado a R de forma muy similar al seudocódigo presentado. De nuevo tendremos el vector de nodos y la lista de arcos con los costes como parámetros de entrada, aunque esta vez no hará falta indicar un nodo de inicio. También aquí hay un paso previo que nos permitirá ahorrar comprobaciones y reducir el tiempo de ejecución. Se encuentra al inicio de la función msTreeKruskal y consiste en ordenar de inicio los arcos de menor a mayor en función de su coste, de forma que al iterar lo hagamos uno a uno, evitando realizar el proceso de búsqueda del arco de menor coste en cada iteración del bucle principal. Dicho bucle se repite hasta que el número de arcos del árbol determina que este es un árbol de expansión. Al finalizar, la función devuelve en una lista el vector de nodos y la matriz con los arcos y los costes del árbol de expansión de coste mínimo encontrado, así como el número de ejecuciones del bucle principal necesitadas para encontrarlo.

<http://eio.usc.es/pub/mte/descargas/ProyectosFinMaster/Proyecto_1075.pdf>