

Obtención del Árbol Abarcador de Coste Mínimo.

Objetivo.

- Aprender a implementar el algoritmo de Prim.
- Aprender a implementar el algoritmo de Kruskal.
- Conocer y aprender a utilizar el TAD Disjoint-Set.
- Aprender a implementar una función de comparación personalizada para el algoritmo `std::sort`.

Descripción.

En esta práctica vamos a obtener el árbol abarcador de coste mínimo de un grafo usando los algoritmos de Prim y Kruskal y comparar sus salidas.

La implementación del algoritmo de Prim es directa a partir de la descripción que hemos visto en clase.

Para implementar el algoritmo de Kruskal vamos a utilizar un TAD auxiliar para “colorear” los vértices del grafo. Este TAD es conocido como “Disjoint Set.” [1].

Supongamos que “ds” es un DisjointSet con tamaño el número de vértices del grafo. Inicialmente cada vértice tiene un “color” distinto (tiene su propio conjunto) y esto se hace con `ds.make_set(v.label())` para todo vértice `v` del grafo.

Después, durante el algoritmo, al seleccionar un lado que conecta dos vértices `u--v`, con `ds.find(u.label())` tenemos el “color” del vértice `u` y con `ds.find(v.label())` el “color” de `v`. Si sus colores son iguales, ese lado no nos interesará, pero si son distintos, ese lado pertenece al AAM y deberemos unir los conjuntos para formar uno solo con `ds.join(u.label(), v.label())`.

Al final del algoritmo, todos los vértices deberían tener el mismo color. En caso contrario es porque el grafo no es conexo.

Por otra parte, para ordenar la lista de lados, vamos a utilizar el algoritmo `std::sort` al que le daremos un funcional (en [2] tienes un ejemplo) para que compare dos lados de forma que la lista quede ordenada de menor a mayor peso y, a igualdad de pesos, $(a,b) < (b,c) < (b,d)$ donde `a`, `b`, `c` y `d` son las etiquetas de los vértices.

Es interesante que compruebes que aunque se pueden obtener soluciones diferentes con Prim (usando distintos nodos origen) y Kruskal, todas tiene el mismo coste mínimo.

Referencias.

[1] https://en.wikipedia.org/wiki/Disjoint-set_data_structure.

[2] <http://www.cplusplus.com/reference/algorithm/sort/>.

