



Práctica de Laboratorio 2

Conjuntos disjuntos para representación de grafos

En python representaremos el tipo de datos *disjointSet* con un diccionario.

Ejemplo: $\{ 'A':1, 'B':2, 'C':1 \}$. Los nodos A y C pertenecen al conjunto de vértices identificado con 1, y el nodo B, al conjunto identificado con 2.

Para cada uno de los siguientes apartados, definir las funciones correspondientes.

1. **make_set(lista)**: inicializa una lista de vértices (*lista*) de modo que cada uno de sus elementos pasen a ser conjuntos unitarios. Retorna un *disjointSet*.
2. **find(elem, disjoint_set)**: obtiene el identificador del conjunto de vértices al que pertenece el elemento *elem*.
3. **union(id_1, id_2, disjoint_set)**: dado dos identificadores de conjuntos de vértices (*id_1* e *id_2*), une dichos conjuntos. Retorna la estructura actualizada.
4. **componentes_conexas(grafo_lista)**: dado un grafo en representación de lista, obtiene sus componentes conexas a través de un *disjointSet*.

Ejemplo:

Entrada: $(['a', 'b', 'c', 'd'], [('a', 'b')])$.

Salida: $\{ 'a': 0, 'b': 0, 'c': 1, 'd': 2 \}$.

Puede testear el comportamiento de esta función utilizando el archivo de test *test_practica2.py*, simplemente corriendo `python3 test_practica2.py`. Agregue todos los casos de prueba que considere pertinentes.

5. **componentes_conexas_lista(grafo_lista)**: dado un grafo en representación de lista, retorna sus componentes conexas. Cada una de ellas es representada como una lista de vértices que la conforman.

Ejemplo:

Entrada: $(['a', 'b', 'c', 'd'], [('a', 'b')])$.

Salida: $[['a', 'b'], ['c'], ['d']]$.