



Apunte de Clases 2

Conjuntos disjuntos para representación de grafos

Tipo abstracto de datos *Union-Find*

El tipo abstracto de datos (TAD) *Union-Find* (también conocido como *DisjointSet*) se utiliza para representar un grupo de conjuntos disjuntos uno a uno. Por lo tanto, estos conjuntos conforman una partición de la unión de los mismos. El TAD debe mantener una correspondencia entre cada uno de los elementos y el subconjunto al que pertenece. Es útil que cada subconjunto tenga un identificador propio. Un Union-Find cuenta con las siguientes operaciones:

- **make_set(xs)**: dada una lista **xs** de elementos, retorna un Union-Find en donde cada elemento de la lista pertenece a un conjunto unitario.
Ejemplo: `make_set([a,b,c,d]) = {{a},{b},{c},{d}}`
- **find_set(x)**: dado un elemento **x**, determina a cuál subconjunto pertenece dicho elemento.
- **union(x,y,unionFind)**: dados dos conjuntos **x** e **y**, actualiza el **unionFind** uniendo ambos conjuntos. Es decir:
$$\text{union}(x,y,\text{unionFind}) = \text{unionFind} \setminus x \setminus y \cup \{x \cup y\}$$

Componentes conexas de un grafo

Se puede utilizar un Union-Find para representar las componentes conexas de un grafo. Los vértices del grafo serán los elementos, y cada componente será un subconjunto. Así, los vértices que pertenecen a una misma componente conexa en el grafo, pertenecerán al mismo subconjunto en el Union-Find.

En la Figura 1 se muestra un ejemplo ilustrativo sobre cómo utilizar las operaciones **make_set** y **union** para representar las componentes conexas de un grafo a través de un Union-Find. La Figura 1 (a) muestra el grafo inicial. La Figura 1 (b) representa el resultado de aplicar **make_set** a cada uno de los vértices del grafo. En este punto, cada vértice pertenece a un subconjunto unitario diferente (indicado por los distintos colores). La Figura 1 (c) muestra el estado del Union-Find cuando se aplica **union** a los conjuntos unitarios de los vértices que conforman la arista A. La Figura 1 (d) muestra el estado del Union-Find cuando se aplica **union** a los conjuntos unitarios de los vértices que conforman la arista B. Finalmente, la Figura 1 (e) muestra el estado último del Union-Find, cuando se unen las componentes conexas de los vértices que conforman la arista C.

Implementación en python

El TAD Union-Find se puede implementar en python de diversas maneras. Una de las posibles es a través del uso de un diccionario. Las claves serán los elementos (en este caso, los vértices del grafo) y el valor almacenado para cada clave será el identificador del subconjunto (componente conexa) al que pertenece. Una posible representación para el ejemplo trabajado en la Figura 1 podría ser:

```
{1: 'verde', 2: 'verde', 3: 'verde', 4: 'azul', 5: 'azul', 6: 'naranja'}
```

En este caso, se utilizan números para representar los vértices del grafo, y nombres de colores para identificar las componentes conexas.

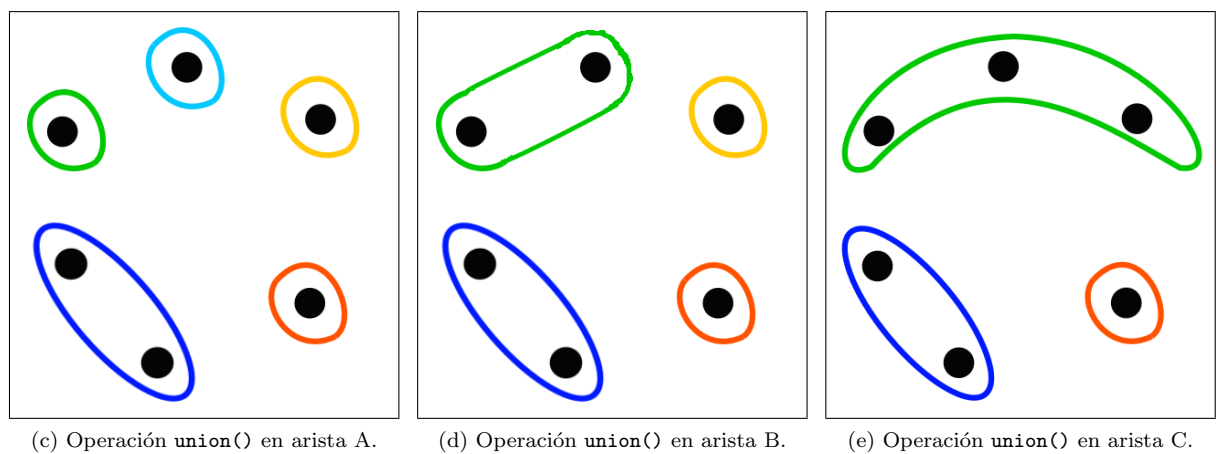
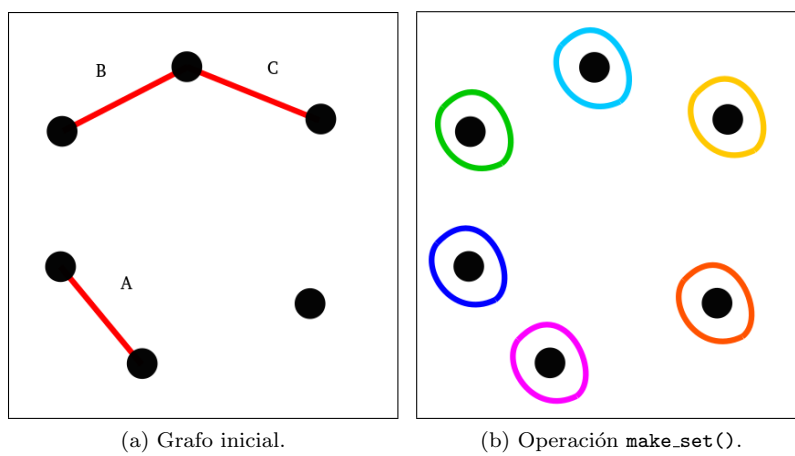


Figura 1: Uso de la estructura de datos Union-Find para representar las componentes de grafos.