

Unir-encontrar

Alberto Benavides
9 de abril de 2020

1. INTRODUCCIÓN

Las operaciones de *unir-encontrar* permiten realizar funciones de búsqueda y unión en conjuntos que tengan elementos con identificadores irrepetibles entre sí. Una manera de representar conjuntos y sus elementos es mediante grafos en que cada nodo corresponde a cada elemento, mientras que grafos disjuntos equivalen a conjuntos, lo cual aparece representado en los incisos de la figura 1.1.

En este caso, la unión de dos conjuntos se realiza mediante la función de búsqueda primeramente. Esta última función consiste en devolver todos los elementos pertenecientes a un conjunto. Una manera de implementar esto es construir un diccionario en el que a cada elemento se le asigne un índice $i = \{0, 1, 2, \dots\}$ y se asocie con el índice p un nodo de otro grafo al que se haya unido. A este otro nodo se le denomina *padre*, por

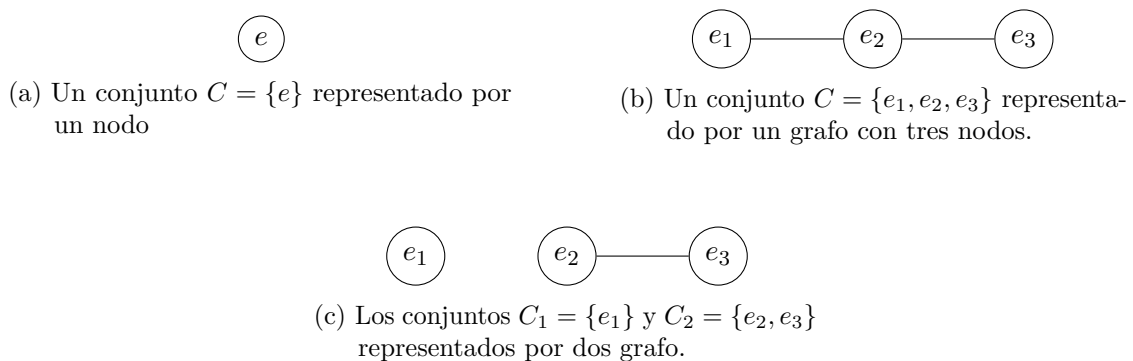


Figura 1.1: Ejemplos de representación de conjuntos y elementos por grafos.

Tabla 1.1: Ejemplo de asignación de índices a elementos de un conjunto.

Elemento	e_1	e_1	e_1	...
Padre	0	1	2	...
Índice	0	1	2	...

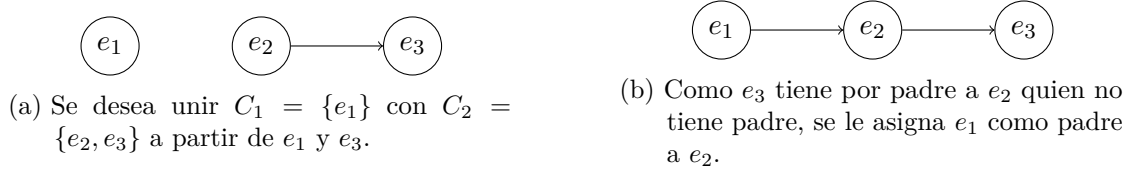


Figura 1.2: Ejemplo de unión de un conjunto $C_1 = \{e_1\}$ con un conjunto $C_2 = \{e_2, e_3\}$.

comodidad. Cuando un elemento se tiene por padre a sí mismo se considera que forma parte de un conjunto con un solo elemento. Esto se ejemplifica en la tabla 1.1.

Unir dos conjuntos a partir de los elementos $e_1 \in C_1$ y $e_2 \in C_2$ consiste en definir qué nodo será padre del otro, aquí e_1 será padre de e_2 . Luego, se recorren todos los padres del nodo hijo hasta encontrar uno que no tenga ningún padre, lo que puede hacerse en tiempo $\mathcal{O}(n)$. Una vez encontrado el nodo que no tiene padre, se le asigna como padre a e_1 y esto resulta en la unión de dos conjuntos representados por grafos con nodos. Una ilustración al respecto puede consultarse en la imagen 1.2.

Una implementación interactiva de esto puede consultarse en <https://jbenavidesv87.github.io/algoritmos/union-find.html>. La implementación funciona a partir de tres métodos de entrada (véase la figura 1.3). El primero permite crear conjuntos con un elemento único, representados por nodos disjuntos. Los otros dos métodos de entrada permiten especificar el valor del elemento del primer conjunto que desea unirse con el valor del elemento de un segundo conjunto.

Las figuras ?? muestran, en ese orden, el ejemplo de agregado de nodos, unión entre nodos disjuntos y unión de dos conjuntos disjuntos a partir de dos de sus elementos.

REFERENCIAS

Búsqueda en anchura y profundidad en árbol binario

Alberto Benavides

Visualización: [greuler](#).

padre de

# padre	Valor	#

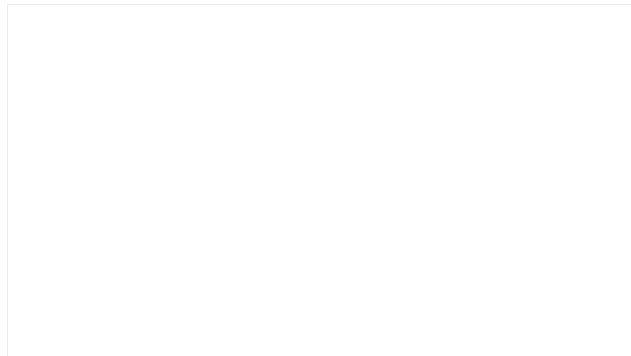


Figura 1.3: Pantalla inicial del interactivo desarrollado.

Búsqueda en anchura y profundidad en árbol binario

Alberto Benavides

Visualización: [greuler](#).

padre de

# padre	Valor	#
0	1	2
1	2	3
0	1	2

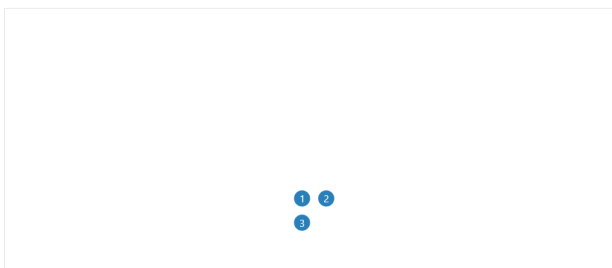


Figura 1.4: Ejemplo de agregado de tres elementos con valores 1, 2, 3, 4.

Búsqueda en anchura y profundidad en árbol binario

Alberto Benavides

Visualización: [greuler](#).

Agregar

padre de

Unir

# Padre	0	0	2	2
Valor	1	2	3	4
#	0	1	2	3

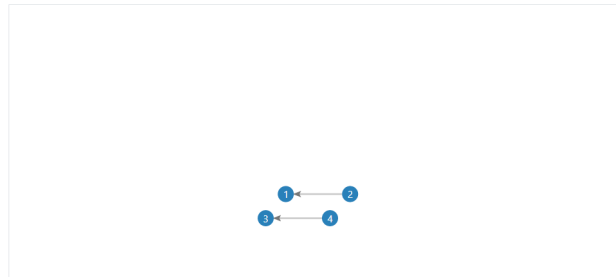


Figura 1.5: Unión del nodo 1 con el 2 y el 3 con el 4.

Búsqueda en anchura y profundidad en árbol binario

Alberto Benavides

Visualización: [greuler](#).

Agregar

padre de

Unir

# Padre	2	0	0	2
Valor	1	2	3	4
#	0	1	2	3

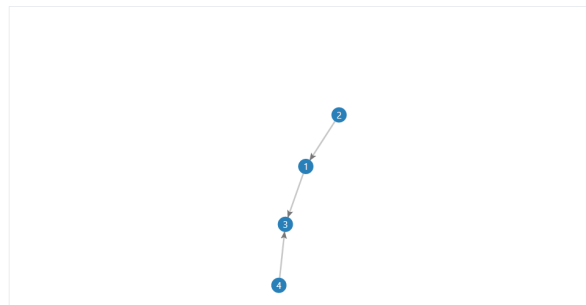


Figura 1.6: Ejemplo de unión del nodo 3 con el 1.