

# Laboratorio 9

José Alejandro Guzmán Zamora

31 de octubre de 2018

## ■ Minimum Spanning Tree

1. En qué caso devolvería el algoritmo de Kruskal diferentes "spanning trees" para el mismo grafo  $G$ ?

El Minimum Spanning Tree que se desarrolla a partir del algoritmo de Kruskal puede ser diferente para ciertos casos. Este algoritmo se basa en ordenar las aristas del grafo en orden no decreciente en base a los pesos de cada uno, por esta razón, cuando hay múltiples aristas con el mismo peso puede ser que (dependiendo del orden) el algoritmo escoga diferentes "minimum spanning trees". Puede ocurrir que las diferentes aristas conecten árboles efectivamente, pero solo si ya se eligió una arista y la siguiente (con el mismo peso) no conecta a otro árbol, no se puede tomar en cuenta.

2. Teniendo un "spanning tree" ya computado. Qué tan rápido se puede encontrar el nuevo "spanning tree" si se agrega un vértice?

A pesar de que el "spanning tree" esté computado, agregar un vértice en el peor de los casos tardaría lo mismo que hacer el "spanning tree" desde el comienzo. La razón principal es que, a pesar de ser solo un vértice, después de agregar el mismo también conlleva considerar una nueva arista con un peso individual, los cambios se pueden ir en cascada dependiendo de los pesos de cada una de las otras aristas.

## ■ Fibonacci Heaps

1.Cuál es la intuición detrás del potencial de las operaciones del Fibonacci Heap?

La función potencial de un Fibonacci Heap se define como  $\phi(H) = t(H) + 2m(H)$  en la que  $t(H)$  es el número de árboles en la lista raíz y  $m(H)$  es la cantidad de nodos marcados en el heap. Lo más interesante de esta función es que dentro de la sumatoria se le duplica al número de nodos marcados. En el caso de la operación de FIB-DECREASE-KEY, es necesario realizar el procedimiento de CASCADING-CUT para quitar la marca de los nodos correspondientes. Al voltear el bit que representa la marca, el potencial de la

estructura se reduce por un valor de 2. De los 2, un potencial cubre el costo de quitar la marca y el otro compensa al potencial general por el movimiento del nodo original a ser parte de la lista raíz.

2. Encontrar el potencial de los siguientes fibonacci heaps.

- a Este heap tiene  $t(H) = 5$  y  $m(H) = 3$ , por lo que  $\phi(H) = 11$
- b Este heap tiene  $t(H) = 3$  y  $m(H) = 3$ , por lo que  $\phi(H) = 9$
- c Este heap tiene  $t(H) = 4$  y  $m(H) = 1$ , por lo que  $\phi(H) = 6$

3. Realizar el trace de la operación Decrease Key sobre este fibonacci heap.

En el caso que se quisiera reducir el key del nodo que tiene valor de 41 (hijo del nodo con key de 38) al valor de 12, lo que pasa es que este nodo pasa a ser parte de la lista raíz y es importante marcar el 38 porque perdió a su hijo. El resultado es el siguiente se puede observar en la figura 1.

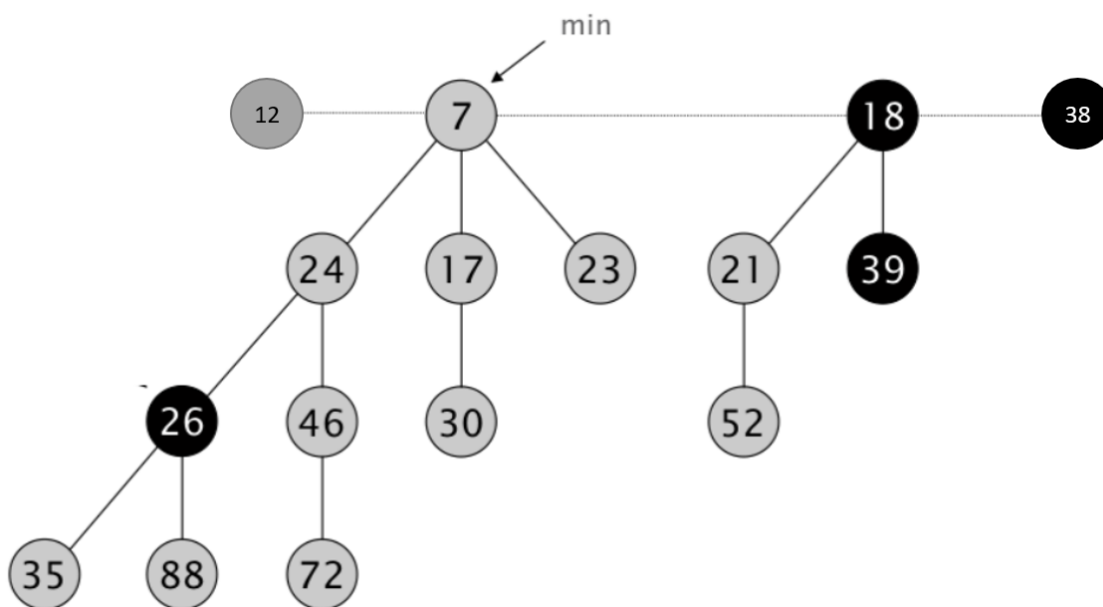


Figura 1: Fibonacci Heap después de Decrease Key en nodo con llave 41.