

# Tarea 08

Marco Antonio Esquivel Basaldua

## Problema 1.

Supongamos que los vértices de cada sub-gráfica fuertemente conectada representan un solo vértice y se crea una arista desde cada uno de los nuevos vértices hacia otro siempre y cuando haya conexión entre ellos. Por ejemplo, la gráfica de la figura 1 se puede interpretar como la gráfica de la figura 2.

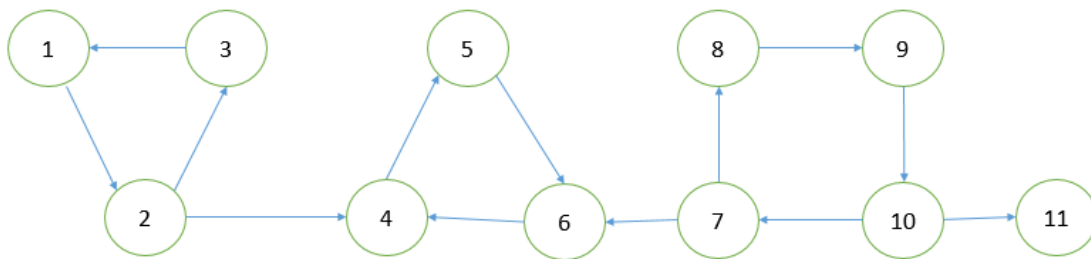


Figura 1

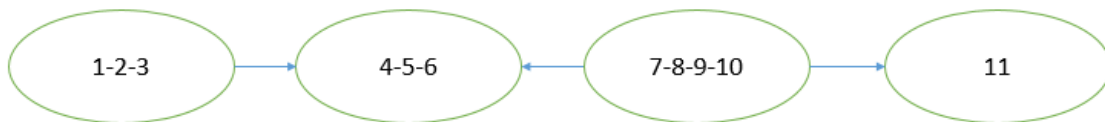


Figura 2

De esta forma se asegura que se tiene una gráfica dirigida acíclica, ya que no hay aristas que formen ciclos entre los nuevos vértices. En caso de que lo anterior se presentara se formaría una sub-gráfica fuertemente conectada pero estas sub-gráficas ya debieron haber sido compactadas a un solo vértice.

Aplicando el algoritmo se realiza el ordenamiento de los vértices basándose según el momento en que cada uno fue recorrido por el DFS, es decir, cuando todos los "hijos" de un vértice fueron visitados ese vértice se pone en la pila. De esta forma se asegura que al menos un vértice (original) dentro de cada sub-gráfica terminará después que la sub-gráfica a la que se va dirigida.

Al aplicar la transpuesta a la gráfica se cambian de dirección las aristas de la figura 2. No es necesario indicar lo mismo en las sub-gráficas dentro de la figura 2 ya que estos vértices siguen estando conectados y se sigue siendo una sub-gráfica fuertemente conectada. Se vuelve a aplicar un DFS y en este caso se asegura que las sub-gráficas (1-2-3) y (7-8-9-10) son exploradas antes que cualquier elemento dentro de las sub-gráficas que apuntan hacia ellas, entonces al seleccionar las sub-gráficas (4-5-6) o (11) se asegura que ya no seguirá explorando fuera de su sub-gráfica.

## Problema 2.

---

### Tiempo.

En la primera fase del algoritmo se realiza el DFS, entonces el tiempo que se toma es  $O(E + V)$ , donde  $E$  es el número de aristas y  $V$  es el número de vértices. Después se transpone la gráfica, lo que toma nuevamente  $O(E + V)$  y por ultimo se recorre la gráfica transpuesta por lo que se toma otra vez  $O(E + V)$ . La complejidad de tiempo es entonces  $3O(E + V)$  lo cual se traduce a  $O(E + V)$ .

### Memoria.

El total de elementos en la pila es el total de elementos en la matriz de adyacencia y en su transpuesta, por tanto la complejidad de la memoria es  $O(V^2)$ .

## Problema 3.

---

En la implementación del algoritmo se trabajan con dos matrices,  $g$  y  $gt$ , que son declaradas de forma global para aprovechar las inicializaciones a cero de todos sus elementos, posteriormente se les da valor 1 a aquellos elementos que indiquen una conexión dirigida entre los vértices de la gráfica. El ejemplo utilizado en la implementación es el mostrado en la figura 1. De igual forma se declaran los arreglos *visitados* y *pila*, que son utilizados para conocer si los elementos ordenados de 1 a  $N$  han sido visitados y la para formar la pila de acuerdo a la primer implementación del algoritmo *DFS*. Nuevamente se hace uso de variables globales y se aprovecha el hecho que sean inicializados en 0.

Posteriormente, se lleva a cabo el algoritmo de implementación haciendo uso de dos funciones, *DFS1* y *DFS2*, que son los encargados de aplicar el algoritmo *DFS* a la matriz de adyacencia y su transpuesta respectivamente.

Como se observa en la figura 2, la cantidad de sub-gráficas fuertemente conectadas es 4.

### Observaciones

Las funciones *DFS1* y *DFS2* utilizadas son sumamente similares por lo que se puede pensar en una forma de dejar una sola funcion que haga distinción en si se está trabajando con la matriz de adyacencia o con su transpuesta.