Grafos dirigidos, DAG y Orden topológico

Colectivo Estructuras de Datos y Algoritmos

Octubre 2021

1. Responda V o F. Justifique en cada caso.
   (a) Si un grafo dirigido G contiene ciclos, el algoritmo $TOPOLOGICAL\_SORT(G)$ visto en conferencia produce una ordenación de los vértices G que minimiza el número de arcos "hacia atrás".
   (b) En un DAG existe al menos un vértice con *indegree* igual a 0 y uno con *outdegree* igual a 0.
   (c) La mayor cantidad de arcos que puede tener un DAG con n vértices es $\theta(n^2)$.
2. Demuestre que un grafo dirigido es un DAG si y solo si existe un orden topológico de sus vértices.
3. Diseñe un algoritmo que dado un grafo dirigido $G = \langle V, E\rangle$ y una lista T que constituye una permutación de sus vértices, permita determinar, en $O(|V| + |E|)$ si T es un orden topológico de G.
4. Demuestre que siendo G un DAG, el siguiente algoritmo devuelve un orden topológico de los vértices de G.

```
T ← lista vacía
while G tenga vértices do
    v ← vértice con indegree igual a 0 en G
    quitar v y todos sus arcos de G
    T.append(v)
end while
return T
```

5. Diseñe un algoritmo que dado un grafo dirigido $G = \langle V, E\rangle$ permita determinar, en $O(|V| + |E|)$ si G es o no un árbol. Se dice que un grafo dirigido es un árbol si su grafo subyacente es un árbol libre y existe en el grafo un nodo (raíz) desde el cual se alcanza al resto de los vértices.
6. Se dice que un grafo dirigido $G = \langle V, E\rangle$ es **semiconexo** si para todo par de vértices $u, v \in V$ se cumple que en G existe un camino de u a v y no de v a u o existe un camino de v a u y no de u a v. Implemente un algoritmo que permita determinar en $O(|V| + |E|)$ si un grafo $G = \langle V, E\rangle$ dado es semiconexo.
7. Sea $G = \langle V, E\rangle$ un grafo dirigido y acíclico y sea k la longitud del camino más largo en G. Diseñe un algoritmo que particione el conjunto de vértices en a lo sumo $k + 1$ grupos de modo que para todo par de vértices u, v con $v \neq u$ pertenecientes a un mismo grupo, no exista camino entre u y v y tampoco exista camino entre v y u. La complejidad temporal de su algoritmo debe ser $O(|V| + |E|)$.
8. Dado un grafo dirigido y acíclico $G = \langle V, E\rangle$ y un vértice $v \in V$, diseñe un algoritmo que determine $\forall u \in V$ la cantidad de caminos que van de v a u. La complejidad temporal de su algoritmo debe ser $O(|V| + |E|)$.
   (a) ¿Cómo se puede modificar su algoritmo para, en lugar de los caminos que comienzan en v, encontrar aquellos que terminan en v?
9. Dado un grafo dirigido y acíclico $G = \langle V, E\rangle$ y un vértice $v \in V$, diseñe un algoritmo que determine $\forall u \in V$ la longitud del camino de longitud máxima que va de v a u. La complejidad temporal de su algoritmo debe ser $O(|V| + |E|)$.
   (a) ¿Cómo se puede modificar su algoritmo para encontrar la longitud del camino de longitud máxima que va de u a v, para todo vértice u?