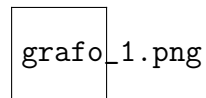


Práctico 1 LENGUAJES: Fundamentos

Mauricio Velasco

1. Implemente una clase `Grafo` que represente un grafo dirigido G como lista de adyacencia. La clase debe recibir sólo el número de vértices del grafo e implementar las operaciones `G.nueva_arista(i,j)`, `G.nuevo_vertice()` y `G.print()`.
 - a) Escriba el código de su implementación.
 - b) Cuánta memoria (como función de n) requiere su clase para representar:
 - 1) Un grafo completo K_n .
 - 2) Un grafo bipartito completo $K_{n,n}$
 - 3) Un ciclo de longitud n .
 - 4) Un árbol con n vértices.
2. Visitemos todos los vértices del grafo no dirigido con $V = \{1, \dots, 7\}$ y con aristas E determinadas por $(1, 2), (1, 3), (3, 7), (3, 6), (2, 4)$ y $(2, 5)$ iniciando en el vértice (2) .
 - a) Escriba la lista de vértices en el orden en el que los visitaríamos en BFS.
 - b) Hay otro orden posible adicional al que escribió en el numeral anterior?
 - c) Cuántos órdenes posibles hay? Escribalos todos.
 - d) Generalizado el ejemplo anterior, cuántos órdenes BFS cree que hay para recorrer un árbol binario con ℓ niveles?.
3. Sea A la matriz de adyacencia de un grafo G . Demuestre que para cualquier entero positivo k y para cualquier par de vértices i, j se tiene que $(A^k)_{ij}$ es igual al número de caminos desde i hasta j de longitud k .
4. Recuerde que un árbol es, por definición, un grafo conexo y sin ciclos. Demuestre las siguientes afirmaciones:

- a) Si G es un grafo con n vértices y m aristas entonces $m = O(n^2)$.
 - b) Todo árbol con n vértices tiene $n - 1$ aristas.
 - c) Concluya que si G es un grafo conexo con m vértices entonces $m = \Omega(n)$ y $m = O(n^2)$.
5. Implemente una función que reciba un **Grafo** H y el índice de un vértice de H y retorne un árbol breadth-first-search T para H . El árbol T debe ser una instancia de la clase **Grafo** del problema (1).
- a) Escriba el código de su implementación.
 - b) Utilice su implementación en el grafo del siguiente dibujo iniciando en el vértice D



y dibuje el árbol T obtenido.