Parcial 2024, 1C: (nota 10)

1)

En un grafo implementado con listas de adyacencia, ¿cuál es la manera más **eficiente** de encontrar un nodo por su clave?

Rta:

Iterar sobre la lista de los nodos.

2)

Si un grafo es **denso** y debemos **recorrerlo frecuentemente**...

Rta:

En general, conviene implementarlo con una matriz de adyacencia. (No eligan la de siempre, es una trampita)

3)

¿Por qué podría resultar útil recorrer un árbol AVL mediante un recorrido **DFS in-orden**?

Rta:

Para ordenar sus datos por clave.

4)

¿Para qué necesitamos **marcas** en los grafos? Puede haber una o más respuestas correctas.

Rtas:

Para que no se produzcan ciclos en el recorrido. Para identificar las componentes conexas de un grafo.

5)

El algoritmo A posee complejidad computacional O(1) , el algoritmo B O(log(n)) , y el algoritmo C O(n^1/2) . ¿Cuál de las siguientes afirmaciones es **verdadera**?

Rta:

Para valores de n grande, C es muchísimo peor que A y B.

6)

¿Cuál es el **propósito principal** de un algoritmo de búsqueda del camino más corto en grafos pesados?

Rta:

Encontrar el camino con el menor peso.

7)

¿Cuál es la **complejidad computacional** del algoritmo floodfill aplicado a un grafo de N nodos y E arcos?

Rta:

O(N + E)

8)

¿**Por qué** se requiere una cola para llevar a cabo el recorrido BFS?

Rta:

Para recordar, en que orden, que nodos, deben visitarse.

9)

¿Cuál de los siguientes tipos de datos **pueden ser eficientes** respecto del consumo de memoria? Puede haber una o varias respuestas correctas, o incluso ninguna.

Rtas:

Un árbol cuyos nodos contienen un arreglo de tamaño fijo con punteros a nodos hijo. El tamaño del arreglo es suficiente para la cantidad máxima de hijos que puede haber. Un árbol cuyos nodos contienen punteros al primer hijo por izquierda y siguiente hermano por derecha.

10)

¿Cuál de los siguientes tipos de datos puede utilizarse para **representar** el organigrama de una empresa? Puede haber una o varias respuestas correctas.

Rtas:

Un árbol cuyos nodos contienen una lista con arreglo de punteros a nodos hijo. Un árbol cuyos nodos contienen una lista doblemente enlazada de punteros a nodos hijo. Un árbol cuyos nodos contienen punteros al primer hijo por izquierda y siguiente hermano por derecha.