

## Examen final regular de Estructura de Datos

1. Dado un grafo dirigido  $G$  y dos rótulos de vértice  $R1$  y  $R2$ . Sea  $V1$  el vértice con rótulo  $R1$  y sea  $V2$  el vértice con rótulo  $R2$ . Usando las operaciones del TDA Grafo escriba en Java un método `RESOLVER(G, R1, R2)` que compute y devuelva en una estructura de datos adecuada todos los caminos simples entre  $V1$  y  $V2$ . Un camino es simple si no tiene vértices repetidos. Si necesita otros TDAs asúmalos como completamente implementados.
2. Diferencias y similitudes entre un árbol 2-3 y un árbol binario de búsqueda. ¿Qué TDAs se implementan con este tipo de árbol? ¿Por qué? Considere aspectos como la definición de la estructura, ubicación de claves, tiempo de operaciones, ejemplos.
3.
  - a) Defina todas las estructuras de datos necesarias para implementar un Árbol General. Programe el constructor de árbol y nodo.
  - b) Agregue un método a la clase árbol General definida anteriormente con la siguiente signatura: `public boolean equals(Tree<E> otro)`. Este método debe implementar una igualdad en profundidad.
  - c) Estimar orden de ejecución del método.
4. ¿Qué es una tabla de hash cerrado? ¿Con que estructuras de datos se implementa? ¿Qué TDAs se pueden implementar con ella? Explique la política de resolución de colisiones lineales. Usar ejemplos, definiciones, tiempos de ejecución, etc.
5.
  - a) Implementar un iterador para una pila implementada con arreglo. (Puede usar otros TDAs).
  - b) Explique HeapSort. Dar tiempo de ejecución. Justificar.